**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

Title:   MPEG-7 Overview (version 8)

Status:  Approved

Source:  Requirements

Editor:  José M. Martínez (UPM-GTI, ES)

# MPEG-7 Overview

## Executive Overview

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the Emmy Award winning standards known as MPEG-1 and MPEG-2, and the MPEG-4 standard. MPEG-1 and MPEG-2 standards made interactive video on CD-ROM and Digital Television possible. MPEG-4 is the multimedia standard for the fixed and mobile web enabling integration of multiple paradigms.

MPEG-7, formally named "Multimedia Content Description Interface", is a standard for describing the multimedia content data that supports some degree of interpretation of the information's meaning, which can be passed onto, or accessed by, a device or a computer code. MPEG-7 is not aimed at any one application in particular; rather, the elements that MPEG-7 standardizes support as broad a range of applications as possible.

More information about MPEG-7 can be found at the MPEG home page (http://mpeg.tilab.com) and the MPEG-7 Alliance website (http://www.mpeg-industry.com). These web pages contain links to a wealth of information about MPEG, including much about MPEG-7, many publicly available documents, several lists of 'Frequently Asked Questions' and links to other MPEG-7 web pages.

This document gives an overview of the MPEG-7 standard, explaining which pieces of technology it includes and what sort of applications are supported by this technology. Also the current work towards MPEG-7 version 2 is presented.

**Table of Contents**

# 1. Introduction

Accessing audio and video used to be a simple matter - simple because of the simplicity of the access mechanisms and because of the poverty of the sources. An incommensurable amount of audiovisual information is becoming available in digital form, in digital archives, on the World Wide Web, in broadcast data streams and in personal and professional databases, and this amount is only growing. The value of information often depends on how easy it can be found, retrieved, accessed and filtered and managed.

The transition between the second and third  millennium abounds with new ways to produce, offer, filter, search, and manage digitized multimedia information. Broadband is being offered with increasing audio and video quality and speed of access.  The trend is clear: in the next few years, users will be confronted with such a large number of contents provided by multiple sources that efficient and accurate access to this almost infinite amount of content seems unimaginable today. In spite of the fact that users have increasing access to these resources, identifying and managing them efficiently is becoming more difficult, because of the sheer volume. This applies to professional as well as end users. The question of identifying and managing content is not just restricted to database retrieval applications such as digital libraries, but extends to areas like broadcast channel selection, multimedia editing, and multimedia directory services.

This challenging situation demands a timely solution to the problem. MPEG-7 is the answer to this need.

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the successful standards known as MPEG-1 (1992) and MPEG-2 (1994), and the MPEG-4 standard (Version 1 in 1998, and version 2 in 1999). The MPEG-1 and MPEG-2 standards have enabled the production of widely adopted commercial products, such as Video CD, MP3, digital audio broadcasting (DAB), DVD, digital television (DVB and ATSC), and many video-on-demand trials and commercial services. MPEG-4 is the first real multimedia representation standard, allowing interactivity and a combination of natural and synthetic material, coded in the form of objects (it models audiovisual data as a composition of these objects). MPEG-4 provides the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the fields of interactive multimedia, mobile multimedia, interactive graphics and enhanced digital television.

The MPEG-7 standard, formally named "Multimedia Content Description Interface", provides a rich set of standardized tools to describe multimedia content. Both human users and automatic systems that process audiovisual information are within the scope of MPEG-7.

MPEG-7 offers a comprehensive set of audiovisual Description Tools (the metadata elements and their structure and relationships, that are defined by the standard in the form of Descriptors and Description Schemes) to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors at the users will), which will form the basis for applications enabling the needed effective and efficient access (search, filtering and browsing) to multimedia content. This is a challenging task given the broad spectrum of requirements and targeted multimedia applications, and the broad number of audiovisual features of importance in such context.

MPEG-7 has been developed by experts representing broadcasters, electronics manufacturers, content creators and managers, publishers and intellectual property rights managers, telecommunication service providers and academia.

More information about MPEG-7 can be found at the MPEG-7 website (http://mpeg.tilab.com) and the MPEG-7 Alliance website (http://www.mpeg-industry.com). These web pages contain links to a wealth of information about MPEG, including much about MPEG-7, many publicly available documents, several lists of 'Frequently Asked Questions' and links to other MPEG-7 web pages.

## 1.1 Context of MPEG-7

More and more audiovisual information is available from many sources around the world. The information may be represented in various forms of media, such as still pictures, graphics, 3D models, audio, speech, video. Audiovisual information plays an important role in our society, be it recorded in such media as film or magnetic tape or originating, in real time, from some audio or visual sensors and be it analogue or, increasingly, digital. While audio and visual information used to be consumed directly by

the human being, there is an increasing number of cases where the audiovisual information is created, exchanged, retrieved, and re-used by computational systems. This may be the case for such scenarios as image understanding (surveillance, intelligent vision, smart cameras, etc.) and media conversion (speech to text, picture to speech, speech to picture, etc.). Other scenarios are information retrieval (quickly and efficiently searching for various types of multimedia documents of interest to the user) and filtering in a stream of audiovisual content description (to receive only those multimedia data items which satisfy the user's preferences). For example, a code in a television program triggers a suitably programmed PVR (Personal Video Recorder) to record that program, or an image sensor triggers an alarm when a certain visual event happens. Automatic transcoding may be performed from a string of characters to audible information or a search may be performed in a stream of audio or video data. In all these examples, the audiovisual information has been suitably "encoded" to enable a device or a computer code to take some action.

Audiovisual sources will play an increasingly pervasive role in our lives, and there will be a growing need to have these sources processed further. This makes it necessary to develop forms of audiovisual information representation that go beyond the simple waveform or sample-based, compression-based (such as MPEG-1 and MPEG-2) or even objects-based (such as MPEG-4) representations. Forms of representation that allow some degree of interpretation of the information's meaning are necessary. These forms can be passed onto, or accessed by, a device or a computer code. In the examples given above an image sensor may produce visual data not in the form of PCM samples (pixels values) but in the form of objects with associated physical measures and time information. These could then be stored and processed to verify if certain programmed conditions are met. A PVR could receive descriptions of the audiovisual information associated to a program that would enable it to record, for example, only news with the exclusion of sport. Products from a company could be described in such a way that a machine could respond to unstructured queries from customers making inquiries.

MPEG-7 is a standard for describing the multimedia content data that will support these operational requirements. The requirements apply, in principle, to both real-time and non real-time as well as push and pull applications. MPEG-7 does not standardize or evaluate applications. In the development of the MPEG-7 standard applications have been used for understanding the requirements and evaluation of technology. It must be made clear that the requirements are derived from analyzing a wide range of potential applications that could use MPEG-7 descriptions. MPEG-7 is not aimed at any one application in particular; rather, the elements that MPEG-7 standardizes support as broad a range of applications as possible.

## 1.2 MPEG-7 Objectives

In October 1996, MPEG started a new work item to provide a solution to the questions described above. The new member of the MPEG family, named "Multimedia Content Description Interface" (in short MPEG-7), provides standardized core technologies allowing description of audiovisual data content in multimedia environments. It extends the limited capabilities of proprietary solutions in identifying content that exist today, notably by including more data types.

Audiovisual data content that has MPEG-7 data associated with it, may include: still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation (scenarios). A special case of these general data types is facial characteristics.

MPEG-7 Description Tools do, however, not depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitised content.

MPEG-7, like the other members of the MPEG family, is a standard representation of audio-visual information satisfying particular requirements. The MPEG-7 standard builds on other (standard) representations such as analogue, PCM, MPEG-1, -2 and -4. One functionality of the MPEG-7 standard is to provide references to suitable portions of them. For example, perhaps a shape descriptor used in MPEG-4 is useful in an MPEG-7 context as well, and the same may apply to motion vector fields used in MPEG-1 and MPEG-2.

MPEG-7 allows different granularity in its descriptions, offering the possibility to have different levels of discrimination. Even though the MPEG-7 description does not depend on the (coded) representation of the material, MPEG-7 can exploit the advantages provided by MPEG-4 coded content. If the material is encoded using MPEG-4, which provides the means to encode audio-visual material as objects having

certain relations in time (synchronisation) and space (on the screen for video, or in the room for audio), it will be possible to attach descriptions to elements (objects) within the scene, such as audio and visual objects.

Because the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications. This implies that the same material can be described using different types of features, tuned to the area of application. To take the example of visual material: a lower abstraction level would be a description of e.g. shape, size, texture, colour, movement (trajectory) and position ('where in the scene can the object be found?'); and for audio: key, mood, tempo, tempo changes, position in sound space. The highest level would give semantic information: 'This is a scene with a barking brown dog on the left and a blue ball that falls down on the right, with the sound of passing cars in the background.' Intermediate levels of abstraction may also exist.

The level of abstraction is related to the way the features can be extracted: many low-level features can be extracted in fully automatic ways, whereas high level features need (much) more human interaction.

Next to having a description of what is depicted in the content, it is also required to include other types of information about the multimedia data:

- The form - An example of the form is the coding format used (e.g. JPEG, MPEG-2), or the overall data size. This information helps determining whether the material can be 'read' by the user's terminal;

- Conditions for accessing the material - This includes links to a registry with intellectual property rights information, and price;

- Classification - This includes parental rating, and content classification into a number of pre-defined categories;

- Links to other relevant material - The information may help the user speeding up the search;

- The context - In the case of recorded non-fiction content, it is very important to know the occasion of the recording (e.g. Olympic Games 1996, final of 200 meter hurdles, men).

In many cases, it is desirable to use textual information for the descriptions. Care was, however, that the usefulness of the descriptions is as independent from the language area as possible. A very clear example where text comes in handy is in giving names of authors, titles, places, etc.

Therefore, MPEG-7 Description Tools allows to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors at the users will) of content that may include:

- Information describing the creation and production processes of the content (director, title, short feature movie).

- Information related to the usage of the content (copyright pointers, usage history, broadcast schedule).

- Information of the storage features of the content (storage format, encoding).

- Structural information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking).

- Information about low level features in the content (colors, textures, sound timbres, melody description).

- Conceptual information of the reality captured by the content (objects and events, interactions among objects).

- Information about how to browse the content in an efficient way (summaries, variations, spatial and frequency subbands, ...).

- Information about collections of objects.

- Information about the interaction of the user with the content (user preferences, usage history).

All these descriptions are of course coded in an efficient way for searching, filtering, etc.

To accommodate this variety of complementary content descriptions, MPEG-7 approaches the description of content from several viewpoints. The sets of Description Tools developed on those viewpoints are presented here as separate entities. However, they are interrelated and can be combined in many ways. Depending on the application, some will present and others can be absent or only partly present.

A description generated using MPEG-7 Description Tools will be associated with the content itself, to allow fast and efficient searching for, and filtering of material that is of interest to the user.

MPEG-7 data may be physically located with the associated AV material, in the same data stream or on the same storage system, but the descriptions could also live somewhere else on the globe. When the content and its descriptions are not co-located, mechanisms that link the multimedia material and their MPEG-7 descriptions are needed; these links will have to work in both directions.

MPEG-7 addresses many different applications in many different environments, which means that it needs to provide a flexible and extensible framework for describing audiovisual data. Therefore, MPEG-7 does not define a monolithic system for content description but rather a set of methods and tools for the different viewpoints of the description of audiovisual content. Having this in mind, MPEG-7 is designed to take into account all the viewpoints under consideration by other leading standards such as, among others, TV Anytime, Dublin Core, SMPTE Metadata Dictionary, and EBU P/Meta. These standardisation activities are focused to more specific applications or application domains, whilst MPEG-7 has been developed as generic as possible. MPEG-7 uses also XML as the language of choice for the textual representation of content description, as XML Schema has been the base for the DDL (Description Definition Language) that is used for the syntactic definition of MPEG-7 Description Tools and for allowing extensibility of Description Tools (either new MPEG-7 ones or application specific). Considering the popularity of XML, usage of it will facilitate interoperability with other metadata standards in the future.

The main elements of the MPEG-7's standard are:

- Description Tools: Descriptors (D), that define the syntax and the semantics of each feature (metadata element); and Description Schemes (DS), that specify the structure and semantics of the relationships between their components, that may be both Descriptors and Description Schemes;

- A Description Definition Language (DDL) to define the syntax of the MPEG-7 Description Tools and to allow the creation of new Description Schemes and, possibly, Descriptors and to allow the extension and modification of existing Description Schemes;

- System tools, to support binary coded representation for efficient storage and transmission, transmission mechanisms (both for textual and binary formats), multiplexing of descriptions, synchronization of descriptions with content, management and protection of intellectual property in MPEG-7 descriptions, etc.

## 1.3 Scope of the Standard

MPEG-7 addresses applications that can be stored (on-line or off-line) or streamed (e.g. broadcast, push models on the Internet), and can operate in both real-time and non real-time environments. A 'real-time environment' in this context means that the description is generated while the content is being captured.

Figure 1 below shows a highly abstract block diagram of a possible MPEG-7 processing chain, included here to explain the scope of the MPEG-7 standard. This chain includes feature extraction (analysis), the description itself, and the search engine (application). To fully exploit the possibilities of MPEG-7 descriptions, automatic extraction of features (or 'Descriptors') will be extremely useful. It is also clear that automatic extraction is not always possible, however. As was noted above, the higher the level of abstraction, the more difficult automatic extraction is, and interactive extraction tools will be of good use. However useful they are, neither automatic nor semi-automatic feature extraction algorithms are inside the scope of the standard. The main reason is that their standardisation is not required to allow interoperability, while leaving space for industry competition. Another reason not to standardise analysis is to allow making good use of the expected improvements in these technical areas.

Also the search engines, filter agents, or any other program that can make use of the description, are not specified within the scope of MPEG-7; again this is not necessary, and here too, competition will produce the best results.
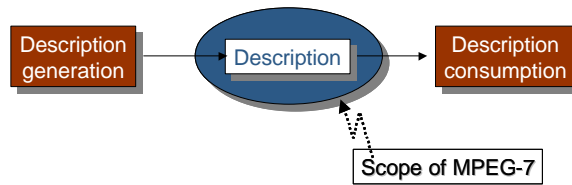
*Figure 1: Scope of MPEG-7*

To provide a better understanding of the terminology introduced above (i.e. Descriptor, Description Scheme, and DDL), please refer to Figure 2 and Figure 3.
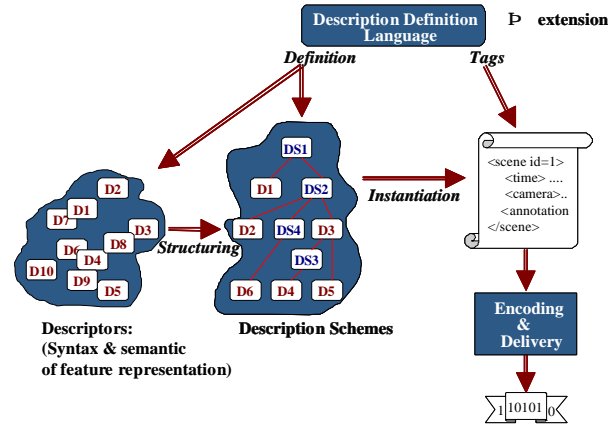


*Figure 2: MPEG-7 main elements*

Figure 2 shows the relationship among the different MPEG-7 elements introduced above. The DDL allows the definition of the MPEG-7 description tools, both Descriptors and Description Schemes, providing the means for structuring the Ds into DSs. The DDL also allows the extension for specific applications of particular DSs. The description tools are instantiated as descriptions in textual format (XML) thanks to the DDL (based on XML Schema). Binary format of descriptions is obtained by means of the BiM defined in the Systems part.
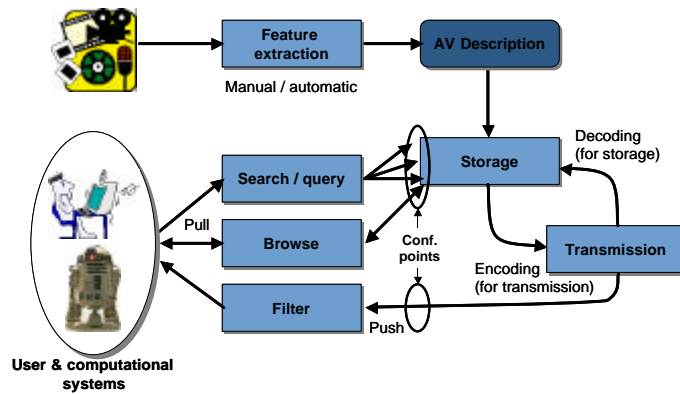
*Figure 3: Abstract representation of possible applications using MPEG-7*

Figure 3 explains a hypothetical MPEG-7 chain in practice[1]. From the multimedia content an Audiovisual description is obtained via manual or semi-automatic extraction. The AV description may be stored (as depicted in the figure) or streamed directly. If we consider a pull scenario, client applications will submit queries to the descriptions repository abd will receive a set of descriptions matching the query for browsing (just for inspecting the description, for manipulating it, for retrieving the described content, etc.). In a push scenario a filter (e.g., an inteligent agent) will select descriptions from the available ones and perform the programmed actions aftyerwards (e.g., switching a broadcast channel or recording the described stream). In both scenarios, the all the modules may handle descriptions coded in MPEG-7 formats (either textual or binary), but only at the indicated conformance points it is required to be MPEG-7 conformant (as they show the interfaces between an application acting as information server and information consumer).

The emphasis of MPEG-7 is the provision of novel solutions for audio-visual content description. Thus, addressing text-only documents was not among the goals of MPEG-7. However, audio-visual content may include or refer to text in addition to its audio-visual information. MPEG-7 therefore has standardized different Description Tools for textual annotation and controlled vocabularies (see below), taking into account existing standards and practices.

Besides the Descriptors themselves, the database structure plays a crucial role in the final retrieval's performance. To allow the desired fast judgement about whether the material is of interest, the indexing information will have to be structured, e.g. in a hierarchical or associative way.

## 1.4 MPEG-7 Application's Areas

The elements that MPEG-7 standardizes provide support to a broad range of applications (for example, multimedia digital libraries, broadcast media selection, multimedia editing, home entertainment devices, etc.). MPEG-7 will also make the web as searchable for multimedia content as it is searchable for text today. This would apply especially to large content archives, which are being made accessible to the public, as well as to multimedia catalogues enabling people to identify content for purchase. The information used for content retrieval may also be used by agents, for the selection and filtering of broadcasted "push" material or for personalized advertising. Additionally, MPEG-7 descriptions will

---

[1] There can be other streams from content to user; these are not depicted here. Furthermore, it is understood that the MPEG-7 Coded Description may be textual or binary, as there might be cases where a binary efficient representation of the description is not needed, and a textual representation would suffice.

allow fast and cost-effective usage of the underlying data, by enabling semi-automatic multimedia presentation and editing.

All application's domains making use of multimedia will benefit from MPEG-7. Considering that at present day it is hard to find one not using multimedia, please extend the list of the examples below using your imagination:

- Architecture, real estate, and interior design (e.g., searching for ideas).

- Broadcast media selection (e.g., radio channel, TV channel).

- Cultural services (history museums, art galleries, etc.).

- Digital libraries (e.g., image catalogue, musical dictionary, bio-medical imaging catalogues, film, video and radio archives).

- E-Commerce (e.g., personalised advertising, on-line catalogues, directories of e-shops).

- Education (e.g., repositories of multimedia courses, multimedia search for support material).

- Home Entertainment (e.g., systems for the management of personal multimedia collections, including manipulation of content, e.g. home video editing, searching a game, karaoke).

- Investigation services (e.g., human characteristics recognition, forensics).

- Journalism (e.g. searching speeches of a certain politician using his name, his voice or his face).

- Multimedia directory services (e.g. yellow pages, Tourist information, Geographical information systems).

- Multimedia editing (e.g., personalised electronic news service, media authoring).

- Remote sensing (e.g., cartography, ecology, natural resources management).

- Shopping (e.g., searching for clothes that you like).

- Social (e.g. dating services).

- Surveillance (e.g., traffic control, surface transportation, non-destructive testing in hostile environments).

The way MPEG-7 data will be used to answer user queries or filtering operations is outside the scope of the standard. The type of content and the query do not have to be the same; for example, visual material may be queried and filtered using visual content, music, speech, etc. It is the responsibility of the search engine and filter agent to match the query data to the MPEG-7 description.

A few query examples are:

- Play a few notes on a keyboard and retrieve a list of musical pieces similar to the required tune, or images matching the notes in a certain way, e.g. in terms of emotions.

- Draw a few lines on a screen and find a set of images containing similar graphics, logos, ideograms,...

- Define objects, including colour patches or textures and retrieve examples among which you select the interesting objects to compose your design.

- On a given set of multimedia objects, describe movements and relations between objects and so search for animations fulfilling the described temporal and spatial relations.

- Describe actions and get a list of scenarios containing such actions.

- Using an excerpt of Pavarotti's voice, obtaining a list of Pavarotti's records, video clips where Pavarotti is singing and photographic material portraying Pavarotti.

## 1.5 Method of Work and Development Schedule

The method of development has been comparable to that of the previous MPEG standards. MPEG work is usually carried out in three stages: definition, competition, and collaboration. In the definition phase, the scope, objectives and requirements for MPEG-7 were defined. In the competitive stage, participants worked on their technology by themselves. The end of this stage was marked by the MPEG-7 Evaluation following an open Call for Proposals (CfP). The Call asked for relevant technology fitting the requirements. In answer to the Call, all interested parties, no matter whether they participate or have participated in MPEG, were invited to submit their technology to MPEG. Some 60 parties submitted, in total, almost 400 proposals, after which MPEG made a fair expert comparison between these submissions.

Selected elements of different proposals were incorporated into a common model (the eXperimentation Model, or XM) during the collaborative phase of the standard with the goalof building the best possible model, which was in essence a draft of the standard itself. During the collaborative phase, the XM was updated and improved in an iterative fashion, until MPEG-7 reached the Committee Draft (CD) stage in October 2000, after several versions of the Working Draft. Improvements to the XM were made through Core Experiments (CEs). CEs were defined to test the existing tools against new contributions and proposals, within the framework of the XM, according to well-defined test conditions and criteria. Finally, those parts of the XM (or of the Working Draft) that corresponded to the normative elements of MPEG-7 were standardized.

The MPEG-7 (version 1) development schedule is shown below:

| | |
|---|---|
| Call for Proposals | October 1998 |
| Evaluation | February 1999 |
| First version of Working Draft (WD) | December 1999 |
| Committee Draft (CD) | October 2000 |
| Final Committee Draft (FCD) | February 2001 |
| Final Draft International Standard (FDIS) | July 2001 |
| International Standard (IS) | September 2001 |

## 1.6 MPEG-7 parts

The MPEG-7 Standard consists of the following parts:

- MPEG-7 Systems - the binary format for encoding MPEG-7 descriptions and the terminal architecture.

- MPEG-7 Description Definition Language - the language for defining the syntax of the MPEG-7 Description Tools and for defining new Description Schemes.

- MPEG-7 Visual – the Description Tools dealing with (only) Visual descriptions.

- MPEG-7 Audio – the Description Tools dealing with (only) Audio descriptions.

- MPEG-7 Multimedia Description Schemes - the Description Tools dealing with generic features and multimedia descriptions.

- MPEG-7 Reference Software - a software implementation of relevant parts of the MPEG-7 Standard with normative status.

- MPEG-7 Conformance Testing - guidelines and procedures for testing conformance of MPEG-7 implementations

- MPEG-7 Extraction and use of descriptions – informative material (in the form of a Technical Report) about the extraction and use of some of the Description Tools (under development).

### 1.7 MPEG Liaisons

MPEG Liaisons deals with organizing formal collaboration between MPEG and other related activities under development in other standardization bodies. Currently MPEG-7 related liaisons include, among others, SMPTE, TV-Anytime, EBU P/Meta, Dublin Core and W3C.

### 1.8 Document structure

This overview document is structured in five sections besides the introduction and several annexes. Each section is divided in several subsections, each of one devoted to the different MPEG-7 parts:

- section 2 describes the major functionalities,

- section 3 contains a detailed technical overview,

- section 4 describes current work regarding profiling within MPEG-7 and

- section 5 describes the work under development, including extensions for version 2.

## 2. Major functionalities in MPEG-7

The following subsections (MPEG-7 part ordered) contain the major functionalities that the different parts of the MPEG-7 Standard offers. The detailed description of the MPEG-7 technologies can be found in section 3.

### 2.1 MPEG-7 Systems

MPEG-7 Systems includes the tools that are needed to prepare MPEG-7 descriptions for efficient transport and storage (binary format), and the terminal architecture and normative interfaces.

### 2.2 MPEG-7 Description Definition Language

According to the definition in the MPEG-7 Requirements Document the Description Definition Language (DDL) is:

"... a language that allows the creation of new Description Schemes and, possibly, Descriptors. It also allows the extension and modification of existing Description Schemes."

The DDL is based on XML Schema Language. But because XML Schema Language has not been designed specifically for audiovisual content description, there are certain MPEG-7 extensions which have been added. As a consequence, the DDL can be broken down into the following logical normative components:

- The XML Schema structural language components;

- The XML Schema datatype language components;

- The MPEG-7 specific extensions.

### 2.3 MPEG-7 Visual

MPEG-7 Visual Description Tools consist of basic structures and Descriptors that cover following basic visual features: color, texture, shape, motion, localization, and face recognition. Each category consists of elementary and sophisticated Descriptors.

### 2.4 MPEG-7 Audio

MPEG-7 Audio provides structures—in conjunction with the Multimedia Description Schemes part of the standard—for describing audio content. Utilizing those structures are a set of low-level Descriptors, for audio features that cut across many applications (e.g., spectral, parametric, and temporal features of a signal), and high-level Description Tools that are more specific to a set of applications. Those high-level

tools include general sound recognition and indexing Description Tools, instrumental timbre Description Tools, spoken content Description Tools, an audio signature Description Scheme, and melodic Description Tools to facilitate query-by-humming.

## 2.5 MPEG-7 Multimedia Description Schemes

MPEG-7 Multimedia Description Schemes (also called MDS) comprises the set of Description Tools (Descriptors and Description Schemes) dealing with generic as well as multimedia entities.

Generic entities are features, which are used in audio and visual descriptions, and therefore "generic" to all media. These are, for instance, "vector", "time", textual description tools, controlled vocabularies, etc.

Apart from this set of generic Description Tools, more complex Description Tools are standardized. They are used whenever more than one medium needs to be described (e.g. audio and video.) These Description Tools can be grouped into 5 different classes according to their functionality:

- Content description: representation of perceivable information
- Content management: information about the media features, the creation and the usage of the AV content;
- Content organization: representation the analysis and classification of several AV contents;
- Navigation and access: specification of summaries and variations of the AV content;
- User interaction: description of user preferences and usage history pertaining to the consumption of the multimedia material.

## 2.6 MPEG-7 Reference Software: the eXperimentation Model

The eXperimentation Model (XM) software is the simulation platform for the MPEG-7 Descriptors (Ds), Description Schemes (DSs), Coding Schemes (CSs), and Description Definition Language (DDL). Besides the normative components, the simulation platform needs also some non-normative components, essentially to execute some procedural code to be executed on the data structures. The data structures and the procedural code together form the applications. The XM applications are divided in two types: the server (extraction) applications and the client (search, filtering and/or transcoding) applications.

## 2.7 MPEG-7 Conformance

MPEG-7 Conformance –under development- will include the guidelines and procedures for testing conformance of MPEG-7 implementations.

## 2.8 MPEG-7 Extraction and use of descriptions

The MPEG-7 Extraction and use of description Technical Report -under development- will include informative material about the extraction and use of some of the Description Tools, both providing additional insight into MPEG-7 Reference Software implementation as well as alternative approaches.

# 3. Detailed technical description of the MPEG-7 Technologies

This section contains a detailed overview of the different MPEG-7 technologies that are curretly standardized. Current developments are described in section 5.

## 3.1 MPEG-7 Multimedia Description Schemes

MPEG-7 Multimedia Description Schemes (DSs) are metadata structures for describing and annotating audio-visual (AV) content. The DSs provide a standardized way of describing in XML the important concepts related to AV content description and content management in order to facilitate searching, indexing, filtering, and access. The DSs are defined using the MPEG-7 Description Definition Language (DDL), which is based on the XML Schema Language, and are instantiated as documents or streams. The

resulting descriptions can be expressed in a textual form (i.e., human readable XML for editing, searching, filtering) or compressed binary form (i.e., for storage or transmission). In this paper, we provide an overview of the MPEG-7 Multimedia DSs and describe their targeted functionality and use in multimedia applications.

The goal of the MPEG-7 standard is to allow interoperable searching, indexing, filtering and access of audio-visual (AV) content by enabling interoperability among devices and applications that deal with AV content description. MPEG-7 describes specific features of AV content as well as information related to AV content management. MPEG-7 descriptions take two possible forms: (1) a textual XML form suitable for editing, searching, and filtering, and (2) a binary form suitable for storage, transmission, and streaming delivery. Overall, the standard specifies four types of normative elements: Descriptors, Description Schemes (DSs), a Description Definition Language (DDL), and coding schemes.

The MPEG-7 Descriptors are designed primarily to describe low-level audio or visual features such as color, texture, motion, audio energy, and so forth, as well as attributes of AV content such as location, time, quality, and so forth. It is expected that most Descriptors for low-level features shall be extracted automatically in applications.

On the other hand, the MPEG-7 DSs are designed primarily to describe higher-level AV features such as regions, segments, objects, events; and other immutable metadata related to creation and production, usage, and so forth. The DSs produce more complex descriptions by integrating together multiple Descriptors and DSs, and by declaring relationships among the description components. In MPEG-7, the DSs are categorized as pertaining to the multimedia, audio, or visual domain. Typically, the multimedia DSs describe content consisting of a combination of audio, visual data, and possibly textual data, whereas, the audio or visual DSs refer specifically to features unique to the audio or visual domain, respectively. In some cases, automatic tools can be used for instantiating the DSs, but in many cases instantiating DSs requires human assisted extraction or authoring tools.

The objective of this section is to provide an overview of the MPEG-7 Multimedia Description SchemesDescription Schemes (DSs) being developed as part of the MPEG-7 standard. The structure of the section is as follows: Section 3.1.1 briefly reviews the organization of the MPEG-7 DSs and highlights the most relevant aspects of the different classes of DSs. Then, Sections 3.1.2 to 3.1.6 describe in more detail the specific design and functionalities of the MPEG-7 Multimedia DSs.

### 3.1.1 Organization of MDS tools

Figure 4 provides an overview of the organization of MPEG-7 Multimedia DSs into the following areas: Basic Elements, Content Description, Content Management, Content Description, Content Organization, Navigation and Access, and User Interaction.
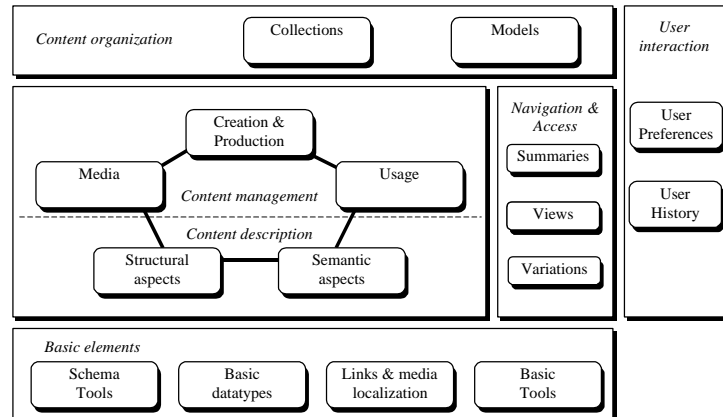


*Figure 4: Overview of the MPEG-7 Multimedia DSs*

### 3.1.1.1 Basic Elements

MPEG-7 provides a number of Schema Tools that assist in the formation, packaging, and annotation of MPEG-7 descriptions. An MPEG-7 description begins with a root element that signifies whether the description is complete or partial. A complete description provides a complete, standalone description of AV content for an application. On the other hand, a description unit carries only partial or incremental information that possibly adds to an existing description. In the case of a complete description, an MPEG-7 top-level element follows the root element. The top-level element orients the description around a specific description task, such as the description of a particular type of AV content, for instance an image, video, audio, or multimedia, or a particular function related to content management, such as creation, usage, summarization, and so forth. The top-level types collect together the appropriate tools for carrying out the specific description task. In the case of description units, the root element can be followed by an arbitrary instance of an MPEG-7 DS or Descriptor. Unlike a complete description which usually contains a "semantically-complete" MPEG-7 description, a description unit can be used to send a partial description as required by an application – such as a description of a place, a shape and texture descriptor and so on. The Package DS describes a user-defined organization of MPEG-7 DSs and Ds into a package, which allows the organized selection of MPEG-7 tools to be communicated to a search engine or user. Furthermore, the DescriptionMetadata DS describes metadata about the description, such as creation time, extraction instrument, version, confidence, and so forth.

A number of basic elements are used throughout the MDS specification as fundamental constructs in defining the MPEG-7 DSs. The basic data types provide a set of extended data types and mathematical structures such as vectors and matrices, which are needed by the DSs for describing AV content. The basic elements include also constructs for linking media files, localizing pieces of content, and describing time, places, persons, individuals, groups, organizations, and other textual annotation. We briefly discuss the MPEG-7 approaches for describing time and textual annotations.
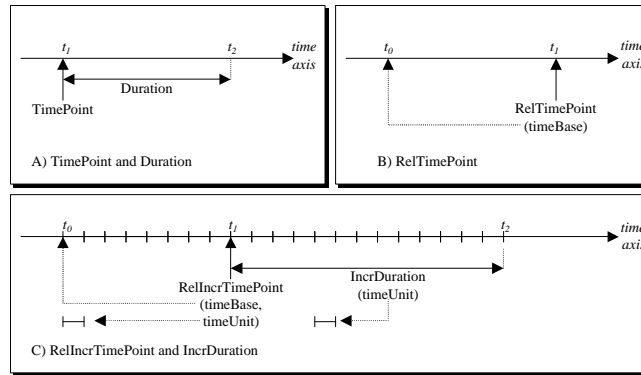


*Figure 5: Overview of the Time DSs*

Temporal Information: the DSs for describing time are based on the ISO 8601 standard, which has also been adopted by the XML Schema language. The Time DS and MediaTime DS describe time information in the real world and in media streams, respectively. Both follow the same strategy described in Figure 5. Figure 5.A illustrates the simplest way to describe a temporal instant and a temporal interval. A time instant, $t_1$, can be described by a lexical representation using the Time Point. An interval, $[t_1, t_2]$, can be described by its starting point, $t_1$, (using the Time Point) and a Duration, $t_2 - t_1$. An alternative way to describe a time instant is shown in Figure 5.B. It relies on Relative Time Point. The instant, $t_1$, is described by a temporal offset with respect to a reference, $t_0$, called Time Base. Note that the goal of the Relative Time Point is to define a temporal instant, $t_1$, and not an interval as the Duration in Figure 5.A. Finally, Figure 5.C illustrates the specification of time using a predefined interval called Time Unit and counting the number of intervals. This specification is particularly efficient for periodic or sampled temporal signals. Since the strategy consists of counting Time Units, the specification of a time instant has to be done relative to a Time Base (or temporal origin). In Figure 5.C, $t_1$ is defined with a Relative Incremental Time Point by counting 8 Time Units (starting from $t_0$). An interval $[t_1, t_2]$, can also be

defined by counting Time Units. In Figure 5.C, Incremental Duration is used to count 13 Time Units to define the interval $[t_1, t_2]$.

Textual Annotation: text annotation is an important component of many DSs. MPEG-7 provides a number of different basic constructs for textual annotation. The most flexible text annotation construct is the data type for free text. Free text allows the formation of an arbitrary string of text, which optionally includes information about the language of the text. However, MPEG-7 provides also a tool for more structured textual annotation by including specific fields corresponding to the questions "Who? What object? What action? Where? When? Why? and How?". Moreover, more complex textual annotations can also be defined by describing explicitly the syntactic dependency between the grammatical elements forming sentences (for example, relation between a verb and a subject, etc.). This last type of textual annotation is particularly useful for applications where the annotation will be processed automatically. Lastly, MPEG-7 provides constructs for classification schemes and controlled terms. The classification schemes provide a language independent set of terms that form a vocabulary for a particular application or domain. Controlled terms are used in descriptions to make reference to the entries in the classification schemes. Allowing controlled terms to be described by classification schemes offers advantages over the standardization of fixed vocabularies for different applications and domains, since it is likely that the vocabularies for multimedia applications will evolve over time.

### 3.1.1.2 Content Management

MPEG-7 provides DSs for AV content management. These tools describe the following information: (1) creation and production, (2) media coding, storage and file formats, and (3) content usage. More details about the MPEG-7 tools for content management are described as follows[2]:

The Creation Information describes the creation and classification of the AV content and of other related materials. The Creation information provides a title (which may itself be textual or another piece of AV content), textual annotation, and information such as creators, creation locations, and dates. The classification information describes how the AV material is classified into categories such as genre, subject, purpose, language, and so forth. It provides also review and guidance information such as age classification, parental guidance, and subjective review. Finally, the Related Material information describes whether there exists other AV materials that are related to the content being described.

The Media Information describes the storage media such as the format, compression, and coding of the AV content. The Media Information DS identifies the master media, which is the original source from which different instances of the AV content are produced. The instances of the AV content are referred to as Media Profiles, which are versions of the master obtained perhaps by using different encodings, or storage and delivery formats. Each Media Profile is described individually in terms of the encoding parameters, storage media information and location.

The Usage Information describes the usage information related to the AV content such as usage rights, usage record, and financial information. The rights information is not explicitly included in the MPEG-7 description, instead, links are provided to the rights holders and other information related to rights management and protection. The Rights DS provides these references in the form of unique identifiers that are under management by external authorities. The underlying strategy is to enable MPEG-7 descriptions to provide access to current rights owner information without dealing with information and negotiation directly. The Usage Record DS and Availability DSs provide information related to the use of the content such as broadcasting, on demand delivery, CD sales, and so forth. Finally, the Financial DS provides information related to the cost of production and the income resulting from content use. The Usage Information is typically dynamic in that it is subject to change during the lifetime of the AV content.

Many of the individual DSs for content management are presented in more detail in section 3.1.2.

### 3.1.1.3 Content Description

MPEG-7 provides DSs for description of the structure and semantics of AV content. The structural tools describe the structure of the AV content in terms of video segments, frames, still and moving regions and

---

[2] Many of the components of the content management DSs are optional. The instantiation of the optional components is often decided in view of the specific multimedia application.

audio segments. The semantic tools describe the objects, events, and notions from the real world that are captured by the AV content.

The functionality of each of these classes of DSs is given as follows:

Structural aspects: describes the audio-visual content from the viewpoint of its structure. The Structure DSs are organized around a Segment DS that represents the spatial, temporal or spatio-temporal structure of the audio-visual content. The Segment DS can be organized into a hierarchical structure to produce a Table of Content for accessing or Index for searching the audio-visual content. The Segments can be further described on the basis of perceptual features using MPEG-7 DescriptorsDescriptors for color, texture, shape, motion, audio features, and so forth, and semantic information using Textual Annotations. The MPEG-7 Structure DSs are further discussed in Section 3.1.3.

Conceptual aspects: describes the audio-visual content from the viewpoint of real-world semantics and conceptual notions. The Semantic DSs involve entities such as objects, events, abstract concepts and relationships. The Structure DSs and Semantic DSs are related by a set of links, which allows the audio-visual content to be described on the basis of both content structure and semantics together. The links relate different Semantic concepts to the instances within the audio-visual content described by the Segments. The MPEG-7 Semantic DSs are further discussed in Section 3.1.3.2.

Many of the individual DSs for content description are presented in more detail in section 3.1.3. Most of the MPEG-7 content description and content management DSs are linked together, and in practice, the DSs are included within each other in MPEG-7 descriptions. For example, Usage information, Creation and Production, and Media information can be attached to individual Segments identified in the MPEG-7 description of audio-visual content structure. Depending on the application, some aspects of the audio-visual content description can be emphasized, such as Semantics or Creation description, while others can be minimized or ignored, such Media or Structure description.

### 3.1.1.4   Navigation and Access

MPEG-7 provides also DSs for facilitating browsing and retrieval of audio-visual content by defining summaries, partitions and decompositions, and variations of the audio-visual material.

Summaries: provide compact summaries of the audio-visual content to enable discovery, browsing, navigation, visualization and sonification of audio-visual content. The Summary DSs involve two types of navigation modes: hierarchical and sequential. In the hierarchical mode, the information is organized into successive levels, each describing the audio-visual content at a different level of detail. In general, the levels closer to the root of the hierarchy provide more coarse summaries, and levels further from the root provide more detailed summaries. The sequential summary provides a sequence of images or video frames, possibly synchronized with audio, which may compose a slide-show or audio-visual skim.

Partitions and Decompositions: describe different decompositions of the audio-visual signals in space, time and frequency. The partitions and decompositions can be used to describe different views of the audio-visual data, which is important for multi-resolution access and progressive retrieval.

Variations: provide information about different variations of audio-visual programs, such as summaries and abstracts; scaled, compressed and low-resolution versions; and versions with different languages and modalities – audio, video, image, text, and so forth. One of the targeted functionalities of the Variation DS is to allow the selection of the most suitable variation of an audio-visual program, which can replace the original, if necessary, to adapt to the different capabilities of terminal devices, network conditions or user preferences.

The Navigation and Access DSs are described in more detail in Section 3.1.4.

### 3.1.1.5   Content Organization

MPEG-7 provides also DSs for organizing and modeling collections of audio-visual content and of descriptions. The Collection DS organizes collections of audio-visual content, segments, events, and/or objects. This allows each collection to be described as a whole based on the common properties. In particular, different models and statistics may be specified for characterizing the attribute values of the collections. The Content Organization DS are described in more detail in Section 3.1.5.

### 3.1.1.6 User Interaction

Finally, the last set of MPEG-7 DSs deals with User Interaction. The User Interaction DSs describe user preferences and usage history pertaining to the consumption of the multimedia material. This allows, for example, matching between user preferences and MPEG-7 content descriptions in order to facilitate personalization of audio-visual content access, presentation and consumption. The main features of the User Interaction DSs are described in Section 3.1.6.

### 3.1.2  Content management

The Content Management Description Tools allow the description of the life cycle of the content, from content to consumption.

The content described by MPEG-7 descriptions can be available in different modalities, formats, Coding Schemes, and there can be several instances. For example, a concert can be recorded in two different modalities: audio and audio-visual. Each of these modalities can be encoded by different Coding Schemes. This creates several media profiles. Finally, several instances of the same encoded content may be available. These concepts of modality, profile and instance are illustrated in Figure 6.



*Figure 6. Model for content, profile and instance*

Content: One reality such as a concert in the world can be represented as several types of media, e.g., audio media, audio-visual media. A content is an entity that has a specific structure to represent the reality.

Media Information: Physical format of a content is described by Media Information DS. One description instance of the DS will be attached to one content entity to describe it. The DS is centered about an identifier for the content entity and it also has sets of descriptorsDescriptors for the storage format of the entity.

Media Profile: One content entity can have one or more media profiles that correspond to different coding schemeCoding Schemes of the entity. One of the profiles is the original one, called master profile, that corresponds to initially created or recorded one. The others will be transcoded from the master. If the content is encoded with the same encoding tool but with different parameters, different media profiles are created.

Media Instance: A content entity can be instantiated as physical entities called media instances. An identifier and a locator specify the media instance.

MPEG-7 Overview                                                                                                          15

CreationInformation: Information about the creation process of a content entity is described by CreationInformation DS. One description instance of the DS will be attached to one content entity to describe it.

UsageInformation: Information about the usage of a content entity is described by UsageInformation DS. One description instance of the DS will be attached to one content entity to describe it.

The only part of the description that depends on the storage media or the encoding format is the MediaInformation described in this section. The remaining part of the MPEG-7 description does not depend on the various profiles or instances and, as a result, can be used to describe jointly all possible copies of the content.

### 3.1.2.1 Media Description Tools

The description of the media involves a single top-level element, the MediaInformation DS. It is composed of an optional MediaIdentification D and one or several MediaProfile Ds

The Media Identification D contains description toolsDescription Tools that are specific to the identification of the AV content, independently of the different available instances. The different media profiles of the content are described via their Media Profile and for each Profile there can be different media instances available.

The Media Profile D contains the different Description Tools that allow the description of one profile of the media AV content being described. The profile concept refers to the different variations that can be produced from an original or master media depending of on the values chosen for the coding, storage format, etc. The profile corresponding to the original or master copy of the AV content is considered the master media profile. For each profile there can be one or more media instances of the master media profile.

The MediaProfile D is composed of:

- MediaFormat D: contains description toolsDescription Tools that are specific to the coding format of the media profile.

- MediaInstance D: contains the description toolsDescription Tools that identify and locate the different media instances (copies) available of a media profile.

- MediaTranscodingHints D: contains description toolsDescription Tools that specify transcoding hints of the media being described. The purpose of this D is to improve quality and reduce complexity for transcoding applications. The transcoding hints can be used in video transcoding and motion estimation architectures to reduce the computational complexity.

- MediaQuality D: represents quality rating information of an audio or visual content. It can be used to represent both subjective quality ratings and objective quality ratings.

### 3.1.2.2 Creation & production Description Tools

The creation and production information Description Tools describe author-generated information about the generation/production process of the AV content. This information cannot usually be extracted from the content itself. This information is related to the material but it is not explicitly depicted in the actual content.

The description of the creation and production information has as top-level element, the CreationInformation DS, which is composed of one Creation D, zero or one Classification D, and zero or several RelatedMaterial Ds.

The Creation D contains the Description Tools related to the creation of the content, including places, dates, actions, materials, staff (technical and artistic) and organizations involved.

The Classification D contains the Description Tools that allow classifying the AV content. The Classification D is used for the description of the classification of the AV content. It allows searching and filtering based on user preferences regarding user-oriented classifications (e.g., language, style, genre, etc.) and service-oriented classifications (e.g., purpose, parental guidance, market segmentation, media review, etc.).

The Related Material D contains the Description Tools related to additional information about the AV content available in other materials.

### 3.1.2.3 Content usage Description Tools

The content usage information Description Tools describe information about the usage process of the AV content.

The description of the usage information is enabled by the UsageInformation DS, which may include one Rights D, zero or one Financial D, and zero or several Availability Ds and UsageRecord Ds.

It is important to note that the UsageInformation DS description may incorporate new descriptions each time the content is used (e.g., UsageRecord DS, Income in Financial datatype), or when there are new ways to access to the content (e.g., Availability D).

The Rights datatype gives access to the information to the rights holders of the annotated content (IPR) and the Access Rights.

The Financial datatype contains information related to the costs generated and income produced by AV content. The notions of partial costs and incomes allows the classification of different costs and incomes as a function of their type. Total and subtotal costs and incomes are to be calculated by the application from these partial values.

The Availability DS contains the Description Tools related to the availability for use of the content.

The UsageRecord DS contains the Description Tools related to the past use of the content.

### 3.1.3    Content Description

### 3.1.3.1 Description of the content structural aspects

The core element of this part of the description is the Segment DS. It addresses the description of the physical and logical aspects of audio-visual content. Segment DSs may be used to form segment trees. MPEG-7 also specifies a Graph DS that allows the representation of complex relations between segments. It is used to describe spatio-temporal relationships, between segments that are not described by the tree structures.

A segment represents a section of an audio-visual content item. The Segment DS is an abstract class (in the sense of object-oriented programming). It has nine major subclasses: Multimedia Segment DS, AudioVisual Region DS, AudioVisual Segment DS, Audio Segment DS, Still Region DS, Still Region 3D DS, Moving Region DS, Video Segment DS and Ink Segment DS. Therefore, it may have both spatial and temporal properties. A temporal segment may be a set of samples in an audio sequence, represented by an Audio Segment DS, a set of frames in a video sequence, represented by a Video Segment DS or a combination of both audio and visual information described by an Audio Visual Segment DS. A spatial segment may be a region in an image or a frame in a video sequence, represented by a Still Region DS for 2D regions and a Still Region 3D DS for 3D regions. A spatio-temporal segment may correspond to a moving region in a video sequence, represented by a Moving Region DS or a more complex combination of visual and audio content for example represented by an AudioVisual Region DS. The InkSegment DS describes a temporal interval or segment of electronic ink data, which corresponds to a set of content ink strokes and/or meta ink strokes. Finally, the most generic segment is the Multimedia Segment DS that describes a composite of segments that form a multimedia presentation. The Segment DS is abstract and cannot be instantiated on its own: it is used to define the common properties of its subclasses. Any segment may be described by creation information, usage information, media information and textual annotation. Moreover, a segment can be decomposed into sub-segments through the Segment Decomposition DS.

The Segment DS describes the result of a spatial, temporal, or spatio-temporal partitioning of the AV content. The Segment DS can describe a recursive or hierarchical decomposition of the AV content into segments that form a segment tree. The SegmentRelation DS describes additional spatio-temporal relationships among segments.

The Segment DS forms the base abstract type of the different specialized segment types: audio segments, video segments, audio-visual segments, moving regions, and still regions. As a result, a segment may

have spatial and/or temporal properties. For example, the AudioSegment DS can describe a temporal audio segment corresponding to a temporal period of an audio sequence. The VideoSegment DS can describe a set of frames of a video sequence. The AudioVisualSegment DS can describe a combination of audio and visual information such as a video with synchronized audio. The StillRegion DS can describe a spatial segment or region of an image or a frame in a video. Finally, the MovingRegion DS can describe a spatio-temporal segment or moving region of a video sequence.

There exists also a set of specialized segments for specific type of AV content. For example, the Mosaic DS is a specialized type of StillRegion. It describes a mosaic or panoramic view of a video segment constructed by aligning together and warping the frames of a VideoSegment upon each other using a common spatial reference system. The VideoText and the InkSegment DSs are two subclasses of the MovingRegion DS. The VideoText DS describes a region of video content corresponding to text or captions. This includes superimposed text as well as text appearing in scene as well as. The InkSegment DS describes a segment of an electronic ink document created by a pen-based system or an electronic whiteboard.

Since the Segment DS is abstract, it cannot be instantiated on its own. However, the Segment DS contains elements and attributes that are common to the different segment types. Among the common properties of segments is information related to creation, usage, media location, and text annotation.

The Segment DS can be used to describe segments that are not necessarily connected, but composed of several non-connected components. Connectivity refers here to both spatial and temporal domains. A temporal segment (Video Segment, Audio Segment and AudioVisual Segment) is said to be temporally connected if it is a sequence of continuous video frames or audio samples. A spatial segment (Still Region) is said spatially connected if it is a group of connected pixels. A spatio-temporal segment (Moving Region) is said spatially and temporally connected if the temporal segment where it is instantiated is temporally connected and if each one of its temporal instantiations in frames is spatially connected (Note that this is not the classical connectivity in a 3D space).
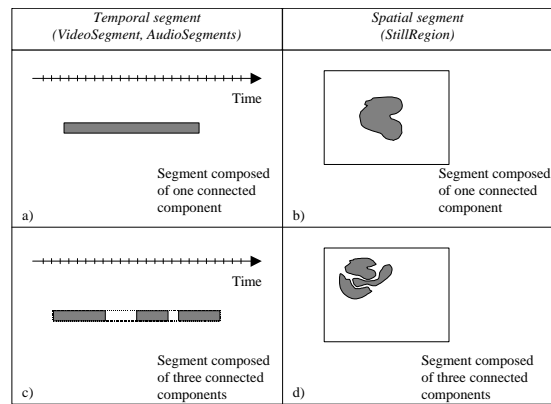


*Figure 7: Examples of segments: a) and b) segments composed of one single connected component; c) and d) segments composed of three connected components*
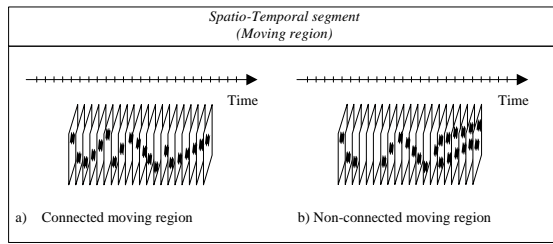
| Spatio-Temporal segment (Moving region) |
|---|
| a) Connected moving region    b) Non-connected moving region |

*Figure 8: Examples of connected a) and non-connected b) moving region*

Figure 7 illustrates several examples of temporal or spatial segments and their connectivity. Figure 7.a) and b) illustrate a temporal and a spatial segment composed of a single connected component. Figure 7.c) and d) illustrate a temporal and a spatial segment composed of three connected components. Figure 8 shows examples of connected and non-connected moving regions. In this last case, the segment is not connected because it is not instantiated in all frames and, furthermore, it involves several spatial connected components in some of the frames.

Note that, in all cases, the Descriptors and DSs attached to the segment are global to the union of the connected components building the segment. At this level, it is not possible to describe individually the connected components of the segment. If connected components have to be described individually, then the segment has to be decomposed into various sub-segments corresponding to its individual connected components.

The Segment DS is recursive, i.e., it may be subdivided into sub-segments, and thus may form a hierarchy (tree). The resulting segment tree is used to describe the media source, the temporal and / or spatial structure of the AV content. For example, a video program may be temporally segmented into various levels of scenes, shots, and micro-segments; a table of contents may thus be generated based on this structure. Similar strategies can be used for spatial and spatio-temporal segments.
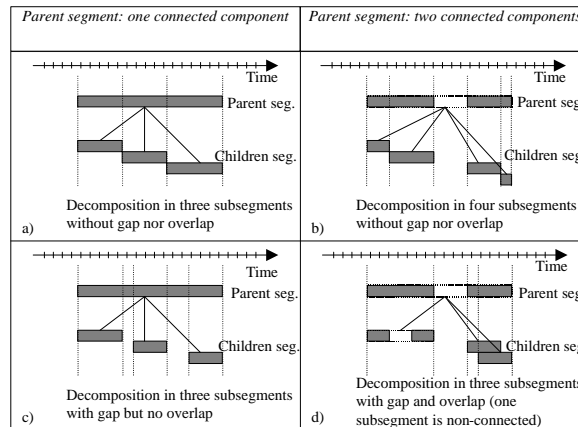


*Figure 9: Examples of Segment Decomposition: a) and b) Segment Decompositions without gap nor overlap; c) and d) Segment Decompositions with gap or overlap.*

A segment may also be decomposed into various media sources such as various audio tracks or viewpoints from several cameras. The hierarchical decomposition is useful to design efficient search strategies (global search to local search). It also allows the description to be scalable: a segment may be described by its direct set of Descriptors and DSs, but it may also be described by the union of the Descriptors and DSs that are related to its sub-segments. Note that a segment may be subdivided into sub-segments of different types, e.g. a video segment may be decomposed in moving regions that are themselves decomposed in still regions.

As it is done in a spatio-temporal space, the decomposition is described by a set of attributes defining the type of sub-division: temporal, spatial or spatio-temporal. Moreover, the spatial and temporal subdivisions may leave gaps and overlaps between the sub-segments. Several examples of decompositions are described for temporal segments in Figure 9. Figure 9.a) and b) describe two examples of decompositions without gaps nor overlaps (partition in the mathematical sense). In both cases the union of the children corresponds exactly to the temporal extension of the parent, even if the parent is itself non connected (see the example of Figure 9b). Figure 9.c) shows an example of decomposition with gaps but no overlaps. Finally, Figure 9.d) illustrates a more complex case where the parent is composed of two connected components and its decomposition creates three children: the first one is itself composed of two connected components, the two remaining children are composed of a single connected component. The decomposition allows gap and overlap. Note that, in any case, the decomposition implies that the union of the spatio-temporal space defined by the children segments is included in the spatio-temporal space defined by their ancestor segment (children are contained in their ancestors).

| Feature | Video Segment | Still Region | Moving Region | Audio Segment |
|---|---|---|---|---|
| Time | X | . | X | X |
| Shape | . | X | X | . |
| Color | X | X | X | . |
| Texture | . | X | . | . |
| Motion | X | . | X | . |
| Camera motion | X | . | . | . |
| Audio features | . | . | X | X |

*Table 1: Specific features for segment description*

As described above, any segment may be described by creation information, usage information, media information and textual annotation. However, specific features depending on the segment type are also allowed. These specific features are reported in Table 1. Most of the Descriptors corresponding to these features can be extracted automatically from the original content. For this purpose, a large number of tools have been reported in the literature. The instantiation of the decomposition involved in the Segment DS can be viewed as a hierarchical segmentation problem where elementary entities (region, video segment, and so forth) have to be defined and structured by inclusion relationship within a tree.

An example of image description is illustrated in Figure 10. The original image is described as a Still Region, SR1, which is described by creation (title, creator), usage information (copyright), media information (file format) as well as a textual annotation (summarizing the image content), a color histogram and a texture descriptor. This initial region can be further decomposed into individual regions. For each decomposition step, we indicate if Gaps and Overlaps are allowed. The segment tree is composed of 8 still regions (note that SR8 is a single segment made of two connected components). For each region, Figure 10 shows the type of feature that is instantiated. Note that it is not necessary to repeat in the tree hierarchy the creation, usage information, and media information, since the children segment are assumed to inherit their parent value (unless re-instantiated).
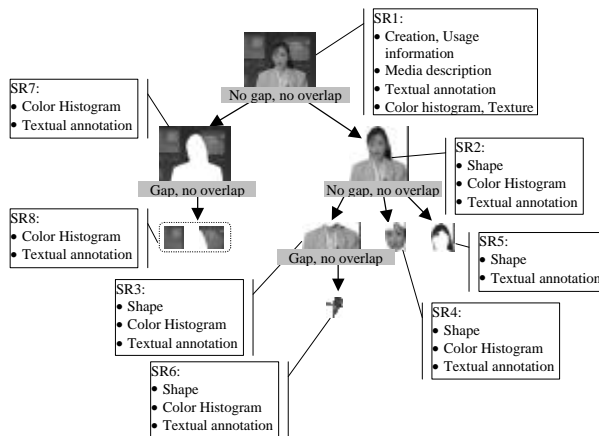
*Figure 10: Examples of Image description with Still Regions.*

The description of the content structure is not constrained to rely on trees. Although, hierarchical structures such as trees are adequate for efficient access, retrieval and scalable description, they imply constraints that may make them inappropriate for certain applications. In such cases, the SegmentRelation DS has to be used. The graph structure is defined very simply by a set of nodes, each corresponding to a segment, and a set of edges, each corresponding to a relationship between two nodes. To illustrate the use of graphs, consider the example shown in Figure 11.
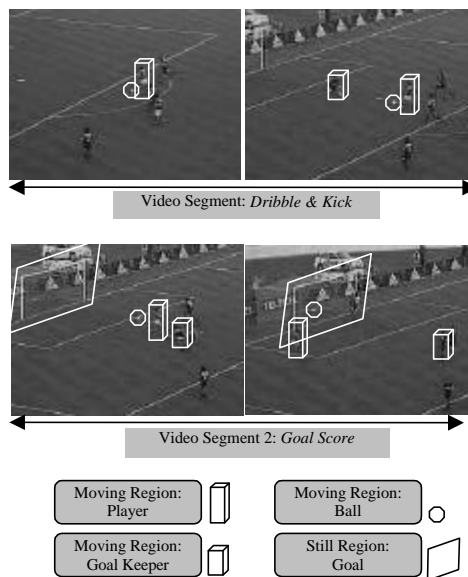


*Figure 11: Example of Video Segments and Regions for the Segment Relationship Graph of Figure 12.*

This example shows an excerpt from a soccer match. Two Video segments, one Still Region and three Moving Regions are considered. A possible graph describing the structure of the content is shown in Figure 12. The Video Segment: Dribble & Kick involves the Ball, the Goalkeeper and the Player. The Ball remains close to the Player who is moving towards the Goalkeeper. The Player appears on the Right of the Goalkeeper. The Goal score video segment involves the same moving regions plus the still region called Goal. In this part of the sequence, the Player is on the Left of the Goalkeeper and the Ball moves towards the Goal. This very simple example illustrates the flexibility of this kind of representation. Note that this description is mainly structural because the relations specified in the graph edges are purely physical and the nodes represent segments (still and moving regions in this example). The only explicit semantic information is available from the textual annotation (where keywords such as Ball, Player, or Goalkeeper can be specified).
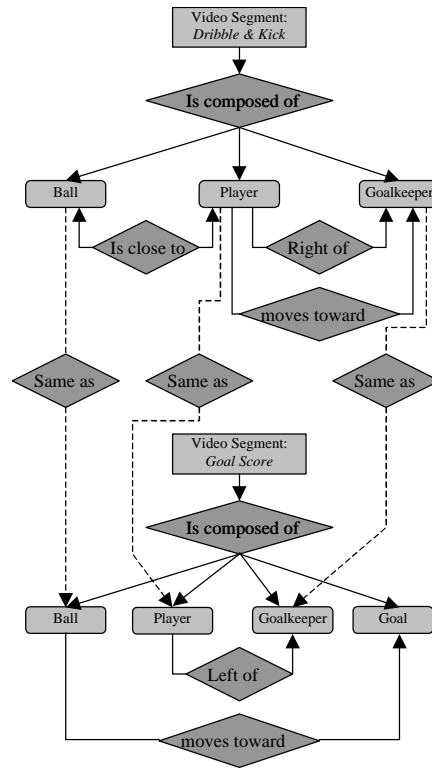


*Figure 12: Example of Segment Relationship Graph.*

### 3.1.3.2 Description of the content conceptual aspects

For some applications, the viewpoint described in the previous section is not appropriate because it highlights the structural aspects of the content. For applications where the structure is of no real use, but where the user is mainly interested in the semantic of the content, an alternative approach is provided by the Semantic DS. In this approach, the emphasis is not on segments but on Events, Objects, Concepts, Places, Time in narrative worlds and Abstraction.

Narrative world refers to the context for a semantic description, that is, it is the "reality" in which the description makes sense. This notion covers the world depicted in the specific instances of audio-visual content as well as more abstract descriptions representing the possible worlds described in the possible media occurrences. A description may involve multiple narrative worlds depicted in multiple instances of AV content.

The SemanticBase DS describes narrative worlds and semantic entities in a narrative world. In addition, a number of specialized DSs are derived from the generic SemanticBase DS, which describe specific types of semantic entities, such as narrative worlds, objects, agent objects, events, places, and time, as follows: The Semantic DS describes narrative worlds that are depicted by or related to the audio-visual content. It may also be used to describe a template for audio-visual content. In practice, the Semantic DS is intended to encapsulate the description of a narrative world. The Object DS describes a perceivable or abstract object. A perceivable object is an entity that exists, i.e. has temporal and spatial extent, in a narrative world (e.g. "Tom's piano"). An abstract object is the result of applying abstraction to a perceivable object (e.g. "any piano"). Essentially, this generates an object template. The AgentObject DS extends from the Object DS. It describes a person, an organization, a group of people, or personalized objects (e.g. "a talking cup in an animated movie"). The Event DS describes a perceivable or abstract event. A perceivable event is a dynamic relation involving one or more objects occurring in a region in time and space of a narrative world (e.g., "Tom playing the piano"). An abstract event is the result of applying abstraction to a perceivable event (e.g. "anyone playing the piano"). Here also, this generates a template of the event. The Concept DS describes a semantic entity that cannot be described as a generalization or abstraction of a specific object, event, time place, or state. It is expressed as a property or collection of properties (e.g. "harmony" or "ripeness"). It may refer to the media directly or to another semantic entity being described. The SemanticState DS describes one or more parametric attributes of a semantic entity at a given time or spatial location in the narrative world, or in a given location in the media (e.g., the piano's weight is 100 kg or the cloudiness of a day). Finally, SemanticPlace and SemanticTime DSs describe respectively a place and a time in a narrative world.

As in the case of the Segment DS, the conceptual aspect of description can be organized in a tree or in a graph. The graph structure is defined by a set of nodes, representing semantic notions, and a set of edges specifying the relationship between the nodes. Edges are described by the Semantic Relation DSs.
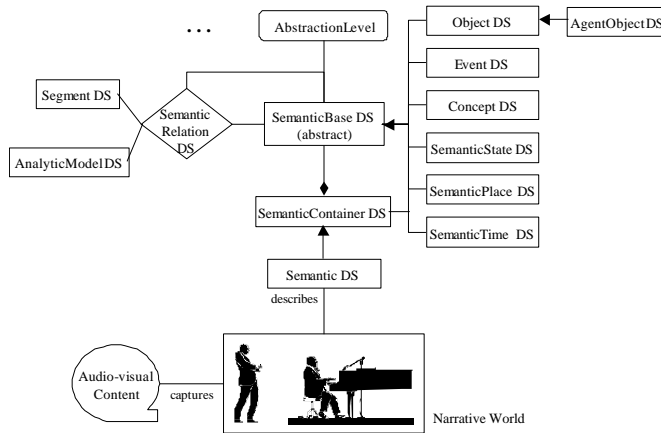


*Figure 13: Tools for the description of conceptual aspects.*

Beside the semantic description of individual instances in audio-visual content, MPEG-7 Semantic DSs also allow the description of abstractions. Abstraction refers to the process of taking a description from a specific instance of audio-visual content and generalizing it to a set of multiple instances of audio-visual content or to a set of specific descriptions. Two types of abstraction, called media abstraction and standard abstraction, are considered.

A media abstraction is a description that has been separated from a specific instance of audio-visual content, and can describe all instances of audio-visual content that are sufficiently similar (similarity depends on the application and on the detail of the description). A typical example is that of a news event, which can be applied to the description of multiple programs, that may have been broadcasted on different channels.

A standard abstraction is the generalization of a media abstraction to describe a general class of semantic entities or descriptions. In general, the standard abstraction is obtained by replacing the specific objects, events or other semantic entities by classes. For instance, if "Tom playing piano" is replaced by "a man playing piano", the description is now a standard abstraction. Standard abstractions can also be recursive, that is one can define abstraction of abstractions. Typically, a standard abstraction is intended for reuse, or to be used by reference in a description.

A simple example of conceptual aspects description is illustrated in Figure 14. The narrative world involves Tom Daniels playing the Piano and his tutor. The event is characterized by a semantic time description: "7-8 PM on the 14$^{th}$ of October 1998", and a semantic place: "Carnegie Hall". The description involves one event: to play, and four objects: piano, Tom Daniels, his tutor and the abstract notion of musicians. The last three objects belong to the class of Agent.
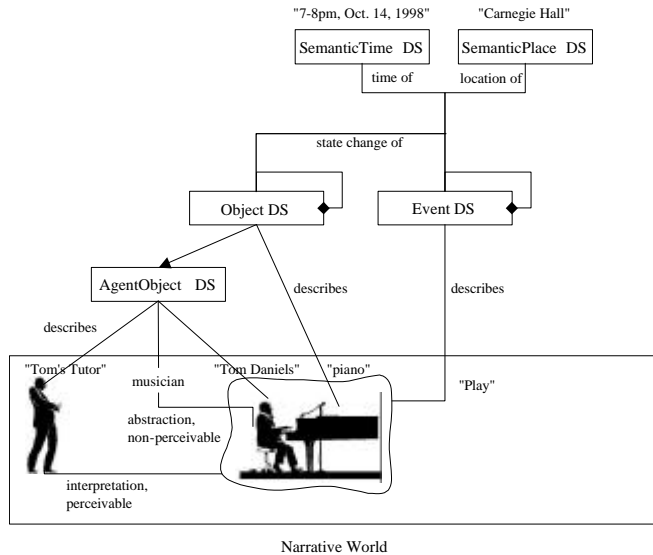


*Figure 14: Example of conceptual aspects description.*

### 3.1.4 Navigation and Access

MPEG-7 facilitates navigation and access of AV content by describing summaries, views and partitions, and variations. The Summary DS describes semantically meaningful summaries and abstracts of AV content in order to enable efficient browsing and navigation. The Space and Frequency View DS describes structural views of the AV signals in the space or frequency domain in order to enable multi-resolution access and progressive retrieval. The Variation DS describes relationships between different variations of AV programs in order to enable adaptive selection under different terminal, delivery, and user preference conditions. These tools are described in more detail as follows:

### *3.1.4.1 Summaries*

The Summarization DS describes different compact summaries of the AV content that facilitate discovery, browsing, and navigation of the AV content. The summary descriptions allow the AV content to be navigated in either a hierarchical or sequential fashion. The hierarchical summary organizes the content into successive levels of detail. The sequential summary composes sequences of images, possibly synchronized with audio, to describe a slide-show or AV skim.

Summarization DS: the MPEG-7 summaries enable fast and effective browsing and navigation by abstracting out the salient information from the AV content. The Summarization DS contains links to the AV content, at the level of segments and frames. Given an MPEG-7 summarization description, a terminal device, such as a digital television set-top box, accesses the AV material composing the summary and renders the result for subsequent interaction with the user. The Summarization DS can describe multiple summaries of the same AV content, such as to provide different levels of detail or highlight specific features, objects, events, or semantics. By including links to the AV content in the summaries, it is possible to generate and store multiple summaries without storing multiple versions of the summary AV content

HierarchicalSummary DS: The HierarchicalSummary DS describes the organization of summaries into multiple levels in order to describe different levels of temporal detail. The HierarchicalSummary DS is constructed around the generic notion of temporal segments of AV content, described by HighlightSegment DSs. Each HighlightSegment contains locators to the AV content that provide access to the associated key-videoclips, key-audioclips, key-frames and key-sounds. Each may also contain textual annotations that describe the key-themes. These audio-visual segments are grouped into summaries, or highlights, using the HighlightSummary DS. For example, in Figure 15, the HierarchicalSummary contains two summaries, where the first summary consists of four highlight segments and the second summary consists of three highlight segments. The summaries could correspond to two different themes and could provide alternative views on the original AV content. The HighlightSummary DS is recursive in nature, enabling summaries to contain other summaries. This capability can be used to build a variety of hierarchical summaries, i.e. to describe content at different granularities. Additionally, multiple summaries may be grouped together using the HierarchicalSummary DS.
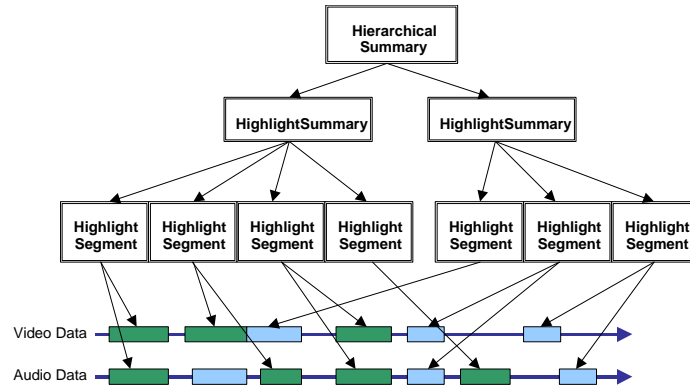


*Figure 15: Illustration of HierarchicalSummary DS containing two summaries*

Figure 16 shows an example of a hierarchical summary of a soccer video. The HierarchicalSummary description gives three levels of detail. In this example, the video of the soccer game is summarized into a single frame at the root. The next level of the hierarchy provides three frames that summarize different

segments of the video. Finally, the bottom level provides additional frames, depicting in more detail the scenes depicted in the segments.
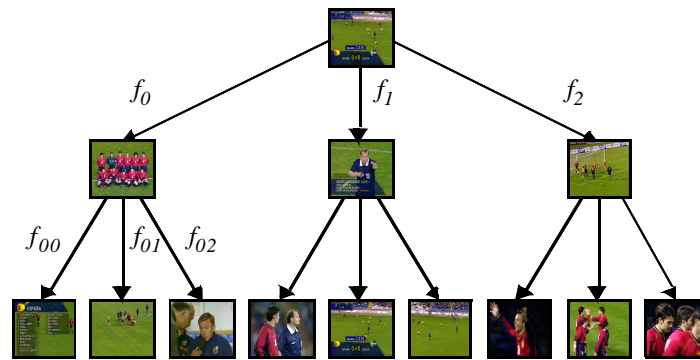


*Figure 16: Example of a Hierarchical Summary of a video of a soccer game providing a multiple level key-frame hierarchy. The Hierarchical Summary denotes the fidelity (i.e., $f_0$, $f_1$) of each key-frame with respect to the video segment referred to by the key-frames at the next lower level.*

SequentialSummary DS: the SequentialSummary DS describes a summary consisting of a sequence of images or video frames, which is possibly synchronized with audio. The SequentialSummary may also contain a sequence of audio clips. The AV content that makes up the SequentialSummary may be stored separately from the original AV content to allow fast navigation and access. Alternatively, the SequentialSummary may link directly to the original AV content in order to reduce storage.

**Comment:** Sentence

### 3.1.4.2 Partitions and Decompositions

The View DS describes a structural view, partition, or decomposition of an audio or visual signal in space, time, and frequency. In general, the views of the signals correspond to low-resolution views, or spatial or temporal segments, or frequency subbands. The Space and Frequency View DS describes a view in terms of its corresponding partition in the space or frequency plane. In addition, the Decomposition DS describes a tree- or graph-based decomposition of an audio or visual signal or organization of views. In the tree- or graph-based decompositions, a node corresponds to a view, and a transition corresponds to an analysis and synthesis signal processing dependency among the connected views.

View DS: the View DS describes a space or frequency view of an audio or visual signal. The SpaceView DS describes a spatial view of an audio or visual signal, for example, a spatial segment of an image. The FrequencyView DS describes a view of an audio or visual signal within a particular frequency band, for example, a wavelet subband of an audio signal. The SpaceFrequencyView DS describes a multi-dimensional view of an audio or visual signal simultaneously in space and frequency, for example, a wavelet subband of a spatial segment of an image. The ResolutionView DS describes a low-resolution view of an audio or visual signal, such as a thumbnail view of an image. Conceptually, a resolution view is a special case of a frequency view that corresponds to a low-frequency subband of the signal. A SpaceResolutionView DS describes a view simultaneously in space and resolution of an audio or visual signal, for example, a low-resolution view of a spatial segment of an image.

View Decompositions: the ViewDecomposition DS describes a space and frequency decomposition or organization of views of an audio or visual signal. The ViewSet DS describes a set of views, which can have different properties of completeness and redundancy. For example, the set of wavelet subbands of an audio signal forms a complete and non-redundant set of views. The SpaceTree DS describes a spatial-tree decomposition of an audio or visual signal, for example, a spatial quad-tree image decomposition. The

FrequencyTree DS describes a frequency-tree decomposition of an audio or visual signal, for example, a wavelet packet-tree image decomposition. The SpaceFrequencyGraph DS describes a decomposition of an audio or visual signal simultaneously in space and frequency in which the views are organized using a space and frequency graph. The VideoViewGraph DS describes a specific type of decomposition of a video signal in both spatial- and temporal-frequency that corresponds to a 3-D subband decomposition. Finally, a MultiResolutionPyramid DS describes a hierarchy of multi-resolution views of an audio or visual signal, such as an image pyramid.
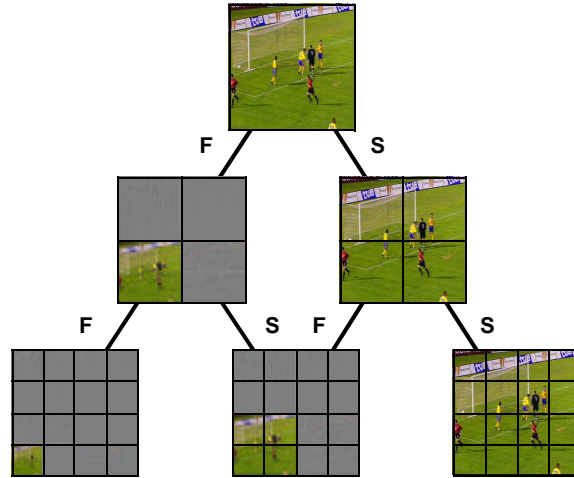


*Figure 17: The Space and Frequency Graph describes the decomposition of an audio or visual signal in space (time) and frequency.*

Figure 17 shows an example Space and Frequency Graph decomposition of an image. The Space and Frequency Graph structure contains nodes that correspond to the different space and frequency views of the image. The views correspond to partitions of the 2-D image signal in space (spatial segments), frequency (wavelet subbands), and space and frequency (wavelet subbands of spatial segments). The space and frequency graph contains also transitions that corresponding to the analysis and synthesis dependencies among the views. For example, in Figure 17, each "S" transition indicates spatial decomposition while each "F" transition indicates frequency or subband decomposition.

### 3.1.4.3   *Variations of the Content*

The Variation DS describes variations of the AV content, such as compressed or low-resolution versions, summaries, different languages, and different modalities, such as audio, video, image, text, and so forth. One of the targeted functionalities of the Variation DS is to allow a server or proxy to select the most suitable variation of the AV content for delivery according to the capabilities of terminal devices, network conditions, or user preferences. The Variations DS describes the different alternative variations. The variations may refer to newly authored AV content, or correspond to AV content derived from another source. A variation fidelity value gives the quality of the variation compared to the original. The variation type attribute indicates the type of variation, such as summary, abstract, extract, modality translation, language translation, color reduction, spatial reduction, rate reduction, compression, and so forth.
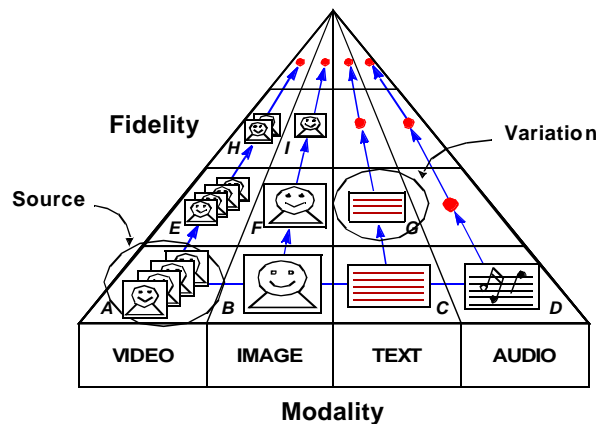
*Figure 18: Illustration of variations of a source AV program.*

Figure 18 illustrates a set of variations of an AV program. The example shows the source video program in the lower left corner (A) and eight variation programs. The variations have different modalities: two variations are video programs (E, H), three are images (B, F, I), two are text C, G), and one is audio (D). Each of the variation programs has a specified fidelity value that indicates the fidelity of the variation program with respect to the source program.

**3.1.5    Content Organization**

MPEG-7 provides DSs for organizing and modeling collections of audio-visual content, segments, events, and/or objects, and describing their common properties. The collections can be further described using different models and statistics in order to characterize the attributes of the collection members.

*3.1.5.1   Collections*

The Collection Structure DS describes collections of audio-visual content or pieces of audio-visual material such as temporal segments of video. The Collection Structure DS groups the audio-visual content, segments, events, or objects into collection clusters and specifies properties that are common to the elements. The CollectionStructure DS describes also statistics and models of the attribute values of the elements, such as a mean color histogram for a collection of images. The CollectionStructure DS also describes relationships among collection clusters.

Figure 19 shows the conceptual organization of the collections within the CollectionStructure DS. In this example, each collection consists of a set of images with common properties, for example, each depicting similar events in a soccer game. Within each collection, the relationships among the images can be specified, such as the degree of similarity of the images in the cluster. Across the collections, the CollectionStructure DS specifies additional relationships, such as the degree of similarity of the collections.
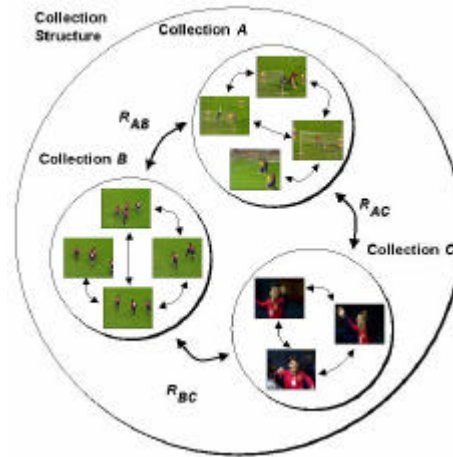
*Figure 19: The Collection Structure DS describes collections of audio-visual content including the relationships (i.e., $R_{AB}$, $R_{BC}$, $R_{AC}$) within and across Collection Clusters.*

### 3.1.5.2  Models

The Model DSs provide tools for modeling the attributes and features of audio-visual content. The Probability Model DS provides fundamental DSs for specifying different statistical functions and probabilistic structures. The Probability Model DS can be used for representing samples of audio-visual data and classes of DescriptorsDescriptors using statistical approximation.

The Analytic Model DSs describe collections of examples of audio-visual data or clusters of DescriptorsDescriptors that provide models for particular semantic classes. The Analytic Model DS specifies semantic labels that indicate the classes being modeled. The Analytic Model DSs also optionally specify the confidence in which the semantic labels are assigned to the models. Further built from the analytic models in MPEG-7 are classifiers. The Classifier DSs describe different types of classifiers that are used to assign semantic labels to audio-visual data based.

### 3.1.6    User Interaction

The UserInteraction DS describe preferences of users pertaining to the consumption of the AV content, as well as usage history. The MPEG-7 AV content descriptions can be matched to the preference descriptions in order to select and personalize AV content for more efficient and effective access, presentation and consumption. The UserPreference DS describes preferences for different types of content and modes of browsing, including context dependency in terms of time and place. The UserPreference DS describes also the weighting of the relative importance of different preferences, the privacy characteristics of the preferences and whether preferences are subject to update, such as by an agent that automatically learns through interaction with the user. The UsageHistory DS describes the history of actions carried out by a user of a multimedia system. The usage history descriptions can be exchanged between consumers, their agents, content providers, and devices, and may in turn be used to determine the user's preferences with regard to AV content.

## 3.2 MPEG-7 Visual

MPEG-7 Visual Description Tools included in the standard consist of basic structures and Descriptors that cover the following basic visual features: Color, Texture, Shape, Motion, Localization, and Face recognition. Each category consists of elementary and sophisticated Descriptors.

### 3.2.1 Basic structures

There are five Visual related Basic structures: the Grid layout, and the Time series, Multiple view, the Spatial 2D coordinates, and Temporal interpolation.

#### 3.2.1.1 Grid layout

The grid layout is a splitting of the image into a set of equally sized rectangular regions, so that each region can be described separately. Each region of the grid can be described in terms of other Descriptors such as color or texture. Furthermore, the descriptor allows to assign the subDescriptors to all rectangular areas, as well as to an arbitrary subset of rectangular regions.

#### 3.2.1.3 Time Series

This descriptor defines a temporal series of Descriptors in a video segment and provides image to video-frame matching and video-frames to video-frames matching functionalities. Two types of TimeSeries are available: RegularTimeSeries and IrregularTimeSeries. In the former, Descriptors locate regularly (with constant intervals) within a given time span. This enables a simple representation for the application that requires low complexity. On the other hand, Descriptors locate irregularly (with various intervals) within a given time span in the latter. This enables an efficient representation for the application that has the requirement of narrow transmission bandwidth or low storage capability. These are useful in particular to build Descriptors that contain time series of Descriptors.

#### 3.2.1.4 2D-3D Multiple View

The 2D/3D Descriptor specifies a structure which combines 2D Descriptors representing a visual feature of a 3D object seen from different view angles. The descriptor forms a complete 3D view-based representation of the object. Any 2D visual descriptor, such as for example contour-shape, region-shape, colour or texture can be used. The 2D/3D descriptor supports integration of the 2D Descriptors used in the image plane to describe features of the 3D (real world) objects. The descriptor allows the matching of 3D objects by comparing their views, as well as comparing pure 2D views to 3D objects.

#### 3.2.1.5 Spatial 2D Coordinates

This description defines a 2D spatial coordinate system and a unit to be used by reference in other Ds/DSs when relevant. The coordinate system is defined by a mapping between an image and the coordinate system. One of the advantages using this descriptor is that MPEG-7 descriptions need not to be modified even if the image size is changed or a part of the image is clipped. In this case, only the description of the mapping from the original image to the edited image is required.

It supports two kinds of coordinate systems: "local" and "integrated" (see Figure 20). In a "local" coordinate system, the coordinates used for the calculation of the description is mapped to the current coordinate system applicable. In an "integrated" coordinate system, each image (frame) of e.g. a video may be mapped to different areas with respect to the first frame of a shot or video. The integrated coordinate system can for instance be used to represent coordinates on a mosaic of a video shot.
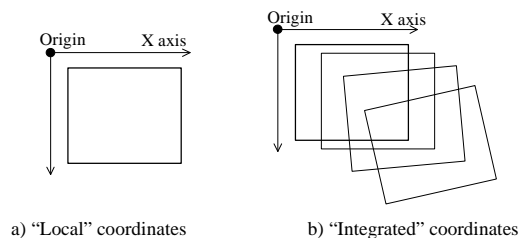


a) "Local" coordinates          b) "Integrated" coordinates

*Figure 20: "local" and "integrated" coordinate system*

### 3.2.1.6 Temporal Interpolation

The TemporalInterpolation D describes a temporal interpolation using connected polynomials. This can be used to approximate multi-dimensional variable values that change with time—such as an object position in a video. The description size of the temporal interpolation is usually much smaller than describing all values. In Figure 21, 25 real values are represented by five linear interpolation functions and two quadratic interpolation functions. The beginning of the temporal interpolation is always aligned to time 0.
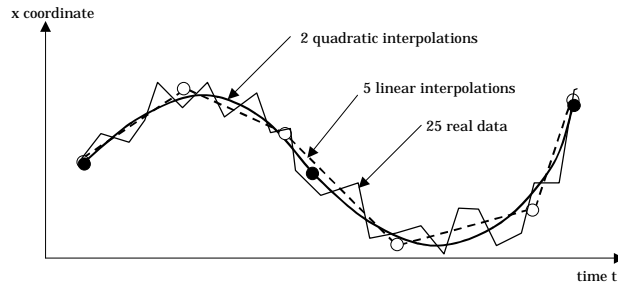


*Figure 21: Real Data and Interpolation functions*

### 3.2.2 Color Descriptors

There are seven Color Descriptors: Color space, Color Quantization, Dominant Colors, Scalable Color, Color Layout, Color-Structure, and GoF/GoP Color.

### 3.2.2.1 Color space

The feature is the color space that is to be used in other color based descriptions.

In the current description, the following color spaces are supported:

- R,G,B

- Y,Cr,Cb

- H,S,V

- HMMD

- Linear transformation matrix with reference to R, G, B

- Monochrome

### 3.2.2.2 Color Quantization

This descriptor defines a uniform quantization of a color space. The number of bins which the quantizer produces is configurable, such that great flexibility is provided for a wide range of applications. For a meaningful application in the context of MPEG-7, this descriptor has to be combined with dominant color descriptors, e.g. to express the meaning of the values of dominant colors.

### 3.2.2.3 Dominant Color(s)

This color descriptor is most suitable for representing local (object or image region) features where a small number of colors are enough to characterize the color information in the region of interest. Whole

images are also applicable, for example, flag images or color trademark images. Color quantization is used to extract a small number of representing colors in each region/image. The percentage of each quantized color in the region is calculated correspondingly. A spatial coherency on the entire descriptor is also defined, and is used in similarity retrieval.

### 3.2.2.4 Scalable Color

The Scalable Color Descriptor is a Color Histogram in HSV Color Space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. The Scalable Color Descriptor is useful for image-to-image matching and retrieval based on color feature. Retrieval accuracy increases with the number of bits used in the representation.

### 3.2.2.5 Color Layout

This descriptor effectively represents the spatial distribution of color of visual signals in a very compact form. This compactness allows visual signal matching functionality with high retrieval efficiency at very small computational costs. It provides image-to-image matching as well as ultra high-speed sequence-to-sequence matching, which requires so many repetitions of similarity calculations. It also provides very friendly user interface using hand-written sketch queries since this descriptors captures the layout information of color feature. The sketch queries are not supported in other color descriptors.

The advantages of this descriptor are:

- that there are no dependency on image/video format, resolutions, and bit-depths. The descriptor can be applied to any still pictures or video frames even though their resolutions are different. It can be also applied both to a whole image and to any connected or unconnected parts of an image with arbitrary shapes.

- that the required hardware/software resources for the descriptor is very small. It needs as law as 8 bytes per image in the default video frame search, and the calculation complexity of both extraction and matching is very low. It is feasible to apply this descriptor to mobile terminal applications where the available resources is strictly limited due to hardware constrain.

- that the captured feature is represented in frequency domain, so that users can easily introduce perceptual sensitivity of human vision system for similarity calculation.

- that it supports scalable representation of the feature by controlling the number of coefficients enclosed in the descriptor. The user can choose any representation granularity depending on their objectives without interoperability problems in measuring the similarity among the descriptors with different granularity. The default number of coefficients is 12 for video frames while 18 coefficients are also recommended for still pictures to achieve a higher accuracy.

### 3.2.2.6 Color-Structure Descriptor

The Color structure descriptor is a color feature descriptor that captures both color content (similar to a color histogram) and information about the structure of this content. Its main functionality is image-to-image matching and its intended use is for still-image retrieval, where an image may consist of either a single rectangular frame or arbitrarily shaped, possibly disconnected, regions. The extraction method embeds color structure information into the descriptor by taking into account all colors in a structuring element of 8x8 pixels that slides over the image, instead of considering each pixel separately. Unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. Color values are represented in the double-coned HMMD color space, which is quantized non-uniformly into 32, 64, 128 or 256 bins. Each bin amplitude value is represented by an 8-bit code. The Color Structure descriptor provides additional functionality and improved similarity-based image retrieval performance for natural images compared to the ordinary color histogram.

### 3.2.2.7 GoF/GoP Color

The Group of Frames/Group of Pictures color descriptor extends the ScalableColor descriptor that is defined for a still image to color description of a video segment or a collection of still images. Additional two bits allows to define how the color histogram was calculated, before the Haar transfor is applied to it: by average, median or intersection. The average histogram, which refers to averaging the counter value of each bin across all frames or pictures, is equivalent to computing the aggregate color histogram of all frames and pictures with proper normalization. The Median Histogram refers to computing the median of the counter value of each bin across all frames or pictures. It is more robust to round-off errors and the presence of outliers in image intensity values compared to the average histogram. The Intersection Histogram refers to computing the minimum of the counter value of each bin across all frames or pictures to capture the "least common" color traits of a group of images. Note that it is different from the histogram intersection, which is a scalar measure. The same similarity/distance measures that are used to compare scalable color descriptions can be employed to compare GoF/GoP color Descriptors.

### 3.2.3 Texture Descriptors

There are three texture Descriptors: Homogeneous Texture, Edge Histogram, and Texture Browsing.

### 3.2.3.1 Homogenous Texture Descriptors

Homogeneous texture has emerged as an important visual primitive for searching and browsing through large collections of similar looking patterns. An image can be considered as a mosaic of homogeneous textures so that these texture features associated with the regions can be used to index the image data. For instance, a user browsing an aerial image database may want to identify all parking lots in the image collection. A parking lot with cars parked at regular intervals is an excellent example of a homogeneous textured pattern when viewed from a distance, such as in an Air Photo. Similarly, agricultural areas and vegetation patches are other examples of homogeneous textures commonly found in aerial and satellite imagery. Examples of queries that could be supported in this context could include "Retrieve all Land-Satellite images of Santa Barbara which have less than 20% cloud cover" or "Find a vegetation patch that looks like this region". To support such image retrieval, an effective representation of texture is required. The Homogeneous Texture Descriptor provides a quantitative representation using 62 numbers (quantified to 8 bits each) that is useful for similarity retrieval. The extraction is done as follows; the image is first filtered with a bank of orientation and scale tuned filters (modeled using Gabor functions) using Gabor filters. The first and the second moments of the energy in the frequency domain in the corresponding sub-bands are then used as the components of the texture descriptor. The number of filters used is 5x6 = 30 where 5 is the number of "scales" and 6 is the number of "directions" used in the multi-resolution decomposition using Gabor functions. An efficient implementation using projections and 1-D filtering operations exists for feature extraction. The Homogeneous Texture descriptor provides a precise quantitative description of a texture that can be used for accurate search and retrieval in this respect. The computation of this descriptor is based on filtering using scale and orientation selective kernels.

### 3.2.3.2 Texture Browsing

The Texture Browsing Descriptor is useful for representing homogeneous texture for browsing type applications, and requires only 12 bits (maximum). It provides a perceptual characterization of texture, similar to a human characterization, in terms of regularity, coarseness and directionality. The computation of this descriptor proceeds similarly as the Homogeneous Texture Descriptor. First, the image is filtered with a bank of orientation and scale tuned filters (modeled using Gabor functions); from the filtered outputs, two dominant texture orientations are identified. Three bits are used to represent each of the dominant orientations. This is followed by analyzing the filtered image projections along the dominant orientations to determine the regularity (quantified to 2 bits) and coarseness (2 bits x 2). The second dominant orientation and second scale feature are optional. This descriptor, combined with the Homogeneous Texture Descriptor, provide a scalable solution to representing homogeneous texture regions in images.

### 3.2.3.3 Edge Histogram

The edge histogram descriptor represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge. Since edges play an important role for image perception,

it can retrieve images with similar semantic meaning. Thus, it primarily targets image-to-image matching (by example or by sketch), especially for natural images with non-uniform edge distribution. In this context, the image retrieval performance can be significantly improved if the edge histogram descriptor is combined with other Descriptors such as the color histogram descriptor. Besides, the best retrieval performances considering this descriptor alone are obtained by using the semi-global and the global histograms generated directly from the edge histogram descriptor as well as the local ones for the matching process.

### 3.2.4 Shape Descriptors

There are three shape Descriptors: Region Shape, Contour Shape, and Shape 3D.

#### 3.2.4.1 Region Shape

The shape of an object may consist of either a single region or a set of regions as well as some holes in the object as illustrated in Figure 22. Since the Region Shape descriptor makes use of all pixels constituting the shape within a frame, it can describe any shapes, i.e. not only a simple shape with a single connected region as in Figure 22 (a) and (b) but also a complex shape that consists of holes in the object or several disjoint regions as illustrated in Figure 22 (c), (d) and (e), respectively. The Region Shape descriptor not only can describe such diverse shapes efficiently in a single descriptor, but is also robust to minor deformation along the boundary of the object.
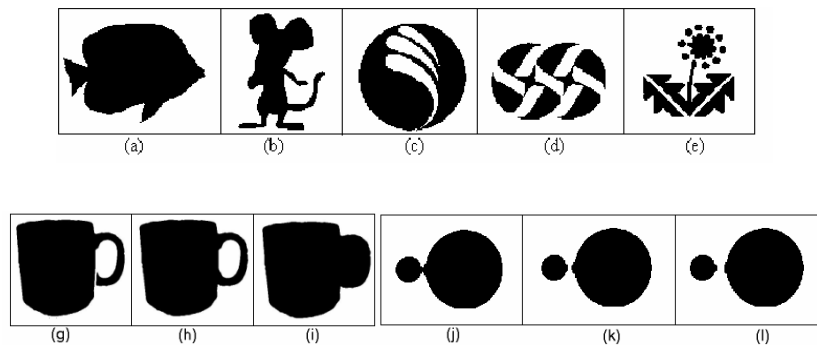


*Figure 22: Examples of various shapes*

Figure 22 (g), (h) and (i) are very similar shape images for a cup. The differences are at the handle. Shape (g) has a crack at the lower handle while the handle in (i) is filled. The region-based shape descriptor considers (g) and (h) similar but different from (i) because the handle is filled. Similarly, Figure 22 (j-l) show the part of video sequence where two disks are being separated. With the region-based descriptor, they are considered similar.

Note that black pixel within the object corresponds to 1 in an image, while white background corresponds to 0.

The descriptor is also characterized by its small size, fast extraction time and matching. The data size for this representation is fixed to 17.5 bytes. The feature extraction and matching processes are straightforward to have low order of computational complexities, and suitable for tracking shapes in the video data processing.

#### 3.2.4.2 Contour Shape

The Contour Shape descriptor captures characteristic shape features of an object or region based on its contour. It uses so-called Curvature Scale-Space representation, which captures perceptually meaningful features of the shape.

The object contour-based shape descriptor is based on the Curvature Scale Space representation of the contour. This representation has a number of important properties, namely:

- It captures very well characteristic features of the shape, enabling similarity-based retrieval

- It reflects properties of the perception of human visual system and offers good generalization

- It is robust to non-rigid motion

- It is robust to partial occlusion of the shape

- It is robust to perspective transformations, which result from the changes of the camera parameters and are common in images and video

- It is compact

Some of the above properties of this descriptor are illustrated in Figure 23, each frame containing very similar images according to CSS, based on the actual retrieval results from the MPEG-7 shape database.
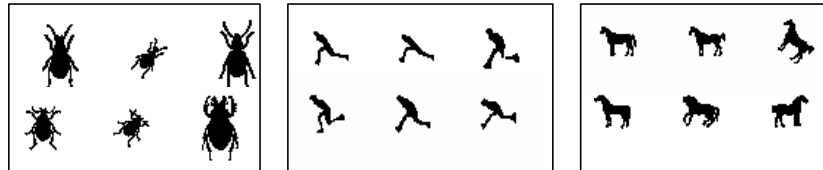


*Figure 23: (a)* shape generalization properties (perceptual similarity among different shapes), (b) robustness to non-rigid motion (man running), (c) robustness to partial occlusion (tails or legs of the horses)

### 3.2.4.3 Shape 3D

Considering the continuous development of multimedia technologies, virtual worlds, and augmented reality, 3D contents become a common feature of today's information systems. Most of the time, 3D information is represented as polygonal meshes. MPEG-4, within the SNHC subgroup, considered this issue and developed technologies for efficient 3D mesh model coding. Within the framework of the MPEG-7 standard, tools for intelligent content-based access to 3D information are needed. The main MPEG-7 applications targeted here are search & retrieval and browsing of 3D model databases.

The 3D Shape Descriptor described in detail provides an intrinsic shape description of 3D mesh models. It exploits some local attributes of the 3D surface.

### 3.2.5　　Motion Descriptors

There are four motion Descriptors: Camera Motion, Motion Trajectory, Parametric Motion, and Motion Activity.

### 3.2.5.1　Camera Motion

This descriptor characterizes 3-D camera motion parameters. It is based on 3-D camera motion parameter information, which can be automatically extracted or generated by capture devices.

The camera motion descriptor supports the following well-known basic camera operations (see Figure 24): fixed, panning (horizontal rotation), tracking (horizontal transverse movement, also called traveling in the film industry), tilting (vertical rotation), booming (vertical transverse movement), zooming (change of the focal length), dollying (translation along the optical axis), and rolling (rotation around the optical axis).
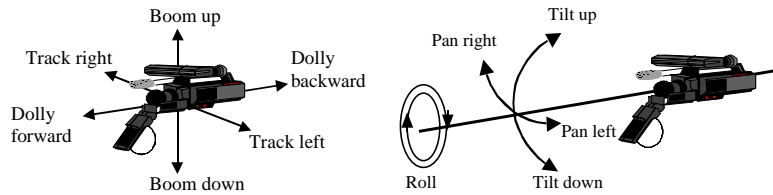
*Figure 24: (a) Camera track, boom, and dolly motion modes, (b) Camera pan, tilt and roll motion modes.*

The sub-shots for which all frames are characterized by a particular type of camera motion, which can be single or mixed, determine the building blocks for the camera motion descriptor. Each building block is described by its start time, the duration, the speed of the induced image motion, by the fraction of time of its duration compared with a given temporal window size, and the focus-of-expansion (FOE) (or focus-of-contraction – FOC). The Descriptor represents the union of these building blocks, and it has the option of describing a mixture of different camera motion types. The mixture mode captures the global information about the camera motion parameters, disregarding detailed temporal information, by jointly describing multiple motion types, even if these motion types occur simultaneously. On the other hand, the non-mixture mode captures the notion of pure motion type and their union within certain time interval. The situations where multiple motion types occur simultaneously are described as a union of the description of pure motion types. In this mode of description, the time window of a particular elementary segment can overlap with time window of another elementary segment.

### 3.2.5.2  Motion Trajectory

The motion trajectory of an object is a simple, high level feature, defined as the localization, in time and space, of one representative point of this object.

This descriptor shows usefulness for content-based retrieval in object-oriented visual databases. It is also of help in more specific applications. In given contexts with a priori knowledge, trajectory can enable many functionalities. In surveillance, alarms can be triggered if some object has a trajectory identified as dangerous (e.g. passing through a forbidden area, being unusually quick, etc.). In sports, specific actions (e.g. tennis rallies taking place at the net) can be recognized. Besides, such a description also allows enhancing data interactions/manipulations: for semiautomatic multimedia editing, trajectory can be stretched, shifted, etc, to adapt the object motion to any given sequence global context.

The descriptor is essentially a list of keypoints (x,y,z,t) along with a set of optional interpolating functions that describe the path of the object between keypoints, in terms of acceleration. The speed is implicitly known by the keypoints specification. The keypoints are specified by their time instant and either their 2-D or 3-D Cartesian coordinates, depending on the intended application. The interpolating functions are defined for each component x(t), y(t), and z(t) independently.

Some of the properties of this representation are that:

- it is independent of the spatio-temporal resolution of the content (e.g., 24 Hz, 30 Hz, 50 Hz, CIF, SIF, SD, HD, etc.), i.e. if the content exists in multiple formats simultaneously, only one set of Descriptors is needed to describe an object trajectory in all instances of that content.

- it is compact and scalable. Instead of storing object coordinate for each frame, the granularity of the descriptor is chosen through the number of keypoints used for each time interval. Besides, interpolating function-data may be discarded, as keypoint-data are already a trajectory description.

- it directly allows wide varieties of uses, like similarity search, or categorization by speed (fast, slow objects), behaviour (accelerating when approaching this area) or by other high level motion characteristics.

### 3.2.5.3 Parametric Motion

Parametric motion models have been extensively used within various related image processing and analysis areas, including motion-based segmentation and estimation, global motion estimation, mosaicing and object tracking. Parametric motion models have been already used in MPEG-4, for global motion estimation and compensation and sprite generation. Within the MPEG-7 framework, motion is a highly relevant feature, related to the spatio-temporal structure of a video and concerning several MPEG-7 specific applications, such as storage and retrieval of video databases and hyperlinking purposes. Motion is also a crucial feature for some domain specific applications that have already been considered within the MPEG-7 framework, such as sign language indexation.

The basic underlying principle consists of describing the motion of objects in video sequences as a 2D parametric model. Specifically, affine models include translations, rotations, scaling and combination of them, planar perspective models make possible to take into account global deformations associated with perspective projections and quadratic models makes it possible to describe more complex movements.

The parametric model is associated with arbitrary (foreground or background) objects, defined as regions (group of pixels) in the image over a specified time interval. In this way, the object motion is captured in a compact manner as a set of a few parameters. Such an approach leads to a very efficient description of several types of motions, including simple translations, rotations and zoomings, or more complex motions such as combinations of the above-mentioned elementary motions.

Defining appropriate similarity measures between motion models is mandatory for effective motion-based object retrieval. It is also necessary for supporting both low level queries, useful in query by example scenarios, and high level queries such as "search for objects approaching the camera", or for "objects describing a rotational motion", or "search for objects translating left", etc.

### 3.2.5.4 Motion Activity

A human watching a video or animation sequence perceives it as being a slow sequence, fast paced sequence, action sequence etc. The activity descriptor captures this intuitive notion of 'intensity of action' or 'pace of action' in a video segment. Examples of high 'activity' include scenes such as 'goal scoring in a soccer match', 'scoring in a basketball game', 'a high speed car chase' etc.. On the other hand scenes such as 'news reader shot', 'an interview scene', 'a still shot' etc. are perceived as low action shots. Video content in general spans the gamut from high to low activity, therefore we need a descriptor that enables us to accurately express the activity of a given video sequence/shot and comprehensively covers the aforementioned gamut. The activity descriptor is useful for applications such as video re-purposing, surveillance, fast browsing, dynamic video summarization, content-based querying etc. For example, we could slow down the presentation frame rate if the activity descriptor indicates high activity so as to make the high activity viewable. Another example of an application is finding all the high action shots in a news video program for example, which can be viewed both as browsing and abstraction.

### 3.2.6 Localization

There two descriptors for localization: Region locator and Spatio-temporal locator

### 3.2.6.1 Region Locator

This descriptor enables localization of regions within images or frames by specifying them with a brief and scalable representation of a Box or a Polygon.

### 3.2.6.2 Spatio Temporal Locator

This describes spatio-temporal regions in a video sequence, such as moving object regions, and provides localization functionality. The main application of it is hypermedia, which displays the related information when the designated point is inside the object. Another main application is object retrieval by checking whether the object has passed particular points. This can be used for surveillance. The SpatioTemporalLocator can describe both spatially connected and non-connected regions.
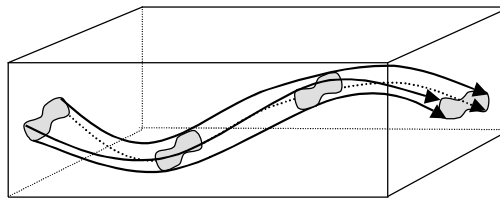
*Figure 25: Spatio-Temporal Region.*

### 3.2.7　Others

#### 3.2.7.1　Face Recognition

The FaceRecognition descriptor can be used to retrieve face images which match a query face image. The descriptor represents the projection of a face vector onto a set of basis vectors which span the space of possible face vectors. The FaceRecognition feature set is extracted from a normalized face image. This normalized face image contains 56 lines with 46 intensity values in each line. The centers of the two eyes in each face image are located on the 24th row and the 16th and 31st column for the right and left eye respectively. This normalized image is then used to extract the one dimensional face vector which consists of the luminance pixel values from the normalized face image arranged into a one dimensional vector using a raster scan starting at the top-left corner of the image and finishing at the bottom-right corner of the image. The FaceRecogniton feature set is then calculated by projecting the one dimensional face vector onto the space defined by a set of basis vectors.

## 3.3 MPEG-7 Audio

MPEG-7 Audio provides structures—building upon some basic structures from the MDS—for describing audio content. Utilizing those structures are a set of low-level Descriptors, for audio features that cut across many applications (e.g., spectral, parametric, and temporal features of a signal), and high-level Description Tools that are more specific to a set of applications. Those high-level tools include the audio signature Description Scheme, musical instrument timbre Description Schemes, the melody Description Tools to aid query-by-humming, general sound recognition and indexing Description Tools, and spoken content Description Tools.

### 3.3.1　MPEG-7 Audio Framework

The Audio Framework contains low-level tools designed to provide a basis for the construction of higher level audio applications. By providing a common platform for the structure of descriptions and the basic semantics for commonly regarded audio features, MPEG-7 Audio establishes a platform for interoperability across all applications that might be built on the framework. The framework provides structures appropriate for representing audio features, and a basic set of features.

#### 3.3.1.1　Structures

There are essentially two ways of describing low-level audio features. One may sample values at regular intervals or one may use Segments (see the discussion on segments in 3.1.1.3) to demark regions of similarity and dissimilarity within the sound. Both of these possibilities are embodied in two low-level descriptor types (one for scalar values, such as power or fundamental frequency, and one for vector types, such as spectra), which create a consistent interface. Any descriptor inheriting from these types can be instantiated, describing a segment with a single summary value or a series of sampled values, as the application requires.

The sampled values themselves may be further manipulated through another unified interface: they can form a Scalable Series. The Scalable Series allows one to progressively down-sample the data contained in a series, as the application, bandwidth, or storage requires. This hierarchical resampling forms a sort of

'scale tree,' which may also store various summary values along the way, such as minimum, maximum, mean, and variance of the descriptor values.

### 3.3.1.2  Features

The low-level audio Descriptors are of general importance in describing audio. There are seventeen temporal and spectral Descriptors that may be used in a variety of applications. They can be roughly divided into the following groups:

- Basic
- Basic Spectral
- Signal Parameters
- Timbral Temporal
- Timbral Spectral
- Spectral Basis

Additionally, a very simple but useful tool is the MPEG-7 silence Descriptor. Each of these classes of audio Descriptors can be seen in Figure 26 and are briefly described below.
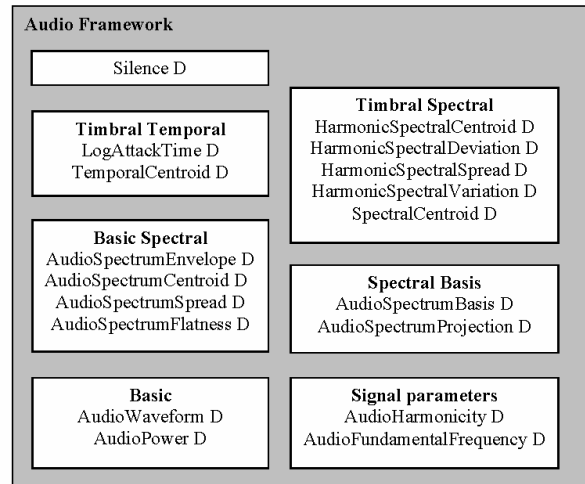


*Figure 26: Overview of Audio Framework including Descriptors*

### 3.3.1.3  Basic

The two basic audio Descriptors are temporally sampled scalar values for general use, applicable to all kinds of signals. The AudioWaveform Descriptor describes the audio waveform envelope (minimum and maximum), typically for display purposes. The AudioPower Descriptor describes the temporally-smoothed instantaneous power, which is useful as a quick summary of a signal, and in conjunction with the power spectrum, below.

### 3.3.1.4  Basic Spectral

The four basic spectral audio Descriptors all share a common basis, all deriving from a single time-frequency analysis of an audio signal. They are all informed by the first Descriptor, the

AudioSpectrumEnvelope Descriptor, which is a logarithmic-frequency spectrum, spaced by a power-of-two divisor or multiple of an octave. This AudioSpectrumEnvelope is a vector that describes the short-term power spectrum of an audio signal. It may be used to display a spectrogram, to synthesize a crude "auralization" of the data, or as a general-purpose descriptor for search and comparison.

The AudioSpectrumCentroid Descriptor describes the center of gravity of the log-frequency power spectrum. This Descriptor is an economical description of the shape of the power spectrum, indicating whether the spectral content of a signal is dominated by high or low frequencies. The AudioSpectrumSpread Descriptor complements the previous Descriptor by describing the second moment of the log-frequency power spectrum, indicating whether the power spectrum is centered near the spectral centroid, or spread out over the spectrum. This may help distinguish between pure-tone and noise-like sounds.

The AudioSpectrumFlatness Descriptor describes the flatness properties of the spectrum of an audio signal for each of a number of frequency bands. When this vector indicates a high deviation from a flat spectral shape for a given band, it may signal the presence of tonal components.

### 3.3.1.5 Signal Parameters

The two signal parameter Descriptors apply chiefly to periodic or quasi-periodic signals. The AudioFundamentalFrequency descriptor describes the fundamental frequency of an audio signal. The representation of this descriptor allows for a confidence measure in recognition of the fact that the various extraction methods, commonly called "pitch-tracking," are not perfectly accurate, and in recognition of the fact that there may be sections of a signal (e.g., noise) for which no fundamental frequency may be extracted. The AudioHarmonicity Descriptor represents the harmonicity of a signal, allowing distinction between sounds with a harmonic spectrum (e.g., musical tones or voiced speech [e.g., vowels]), sounds with an inharmonic spectrum (e.g., metallic or bell-like sounds) and sounds with a non-harmonic spectrum (e.g., noise, unvoiced speech [e.g., fricatives like 'f'], or dense mixtures of instruments).

### 3.3.1.6 Timbral Temporal

The two timbral temporal Descriptors describe temporal characteristics of segments of sounds, and are especially useful for the description of musical timbre (characteristic tone quality independent of pitch and loudness). Because a single scalar value is used to represent the evolution of a sound or an audio segment in time, these Descriptors are not applicable for use with the Scalable Series. The LogAttackTime Descriptor characterizes the "attack" of a sound, the time it takes for the signal to rise from silence to the maximum amplitude. This feature signifies the difference between a sudden and a smooth sound. The TemporalCentroid Descriptor also characterizes the signal envelope, representing where in time the energy of a signal is focused. This Descriptor may, for example, distinguish between a decaying piano note and a sustained organ note, when the lengths and the attacks of the two notes are identical.

### 3.3.1.7 Timbral Spectral

The five timbral spectral Descriptors are spectral features in a linear-frequency space especially applicable to the perception of musical timbre. The SpectralCentroid Descriptor is the power-weighted average of the frequency of the bins in the linear power spectrum. As such, it is very similar to the AudioSpectrumCentroid Descriptor, but specialized for use in distinguishing musical instrument timbres. It is has a high correlation with the perceptual feature of the "sharpness" of a sound.

The four remaining timbral spectral Descriptors operate on the harmonic regularly-spaced components of signals. For this reason, the descriptors are computed in linear-frequency space. The HarmonicSpectralCentroid is the amplitude-weighted mean of the harmonic peaks of the spectrum. It has a similar semantic to the other centroid Descriptors, but applies only to the harmonic (non-noise) parts of the musical tone. The HarmonicSpectralDeviation Descriptor indicates the spectral deviation of log-amplitude components from a global spectral envelope. The HarmonicSpectralSpread describes the amplitude-weighted standard deviation of the harmonic peaks of the spectrum, normalized by the instantaneous HarmonicSpectralCentroid. The HarmonicSpectralVariation Descriptor is the normalized correlation between the amplitude of the harmonic peaks between two subsequent time-slices of the signal.

### 3.3.1.8 Spectral Basis

The two spectral basis Descriptors represent low-dimensional projections of a high-dimensional spectral space to aid compactness and recognition. These descriptors are used primarily with the Sound Classification and Indexing Description Tools, but may be of use with other types of applications as well. The AudioSpectrumBasis Descriptor is a series of (potentially time-varying and/or statistically independent) basis functions that are derived from the singular value decomposition of a normalized power spectrum. The AudioSpectrumProjection Descriptor is used together with the AudioSpectrumBasis Descriptor, and represents low-dimensional features of a spectrum after projection upon a reduced rank basis.

Together, the descriptors may be used to view and to represent compactly the independent subspaces of a spectrogram. Often these independent subspaces (or groups thereof) correlate strongly with different sound sources. Thus one gets more salience and structure out of a spectrogram while using less space. For example, in Figure 27, a pop song is represented by an AudioSpectrumEnvelope Descriptor, and visualized using a spectrogram. The same song has been data-reduced in Figure 28, and yet the individual instruments become more salient in this representation.
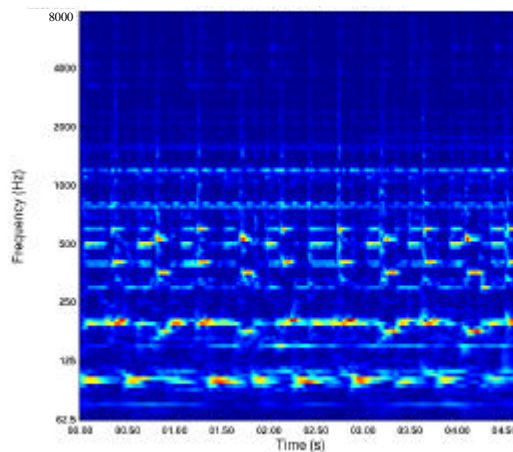


*Figure 27:* `AudioSpectrumEnvelope` *description of a pop song. The required data storage is* NM *values where* N *is the number of spectrum bins and* M *is the number of time points*
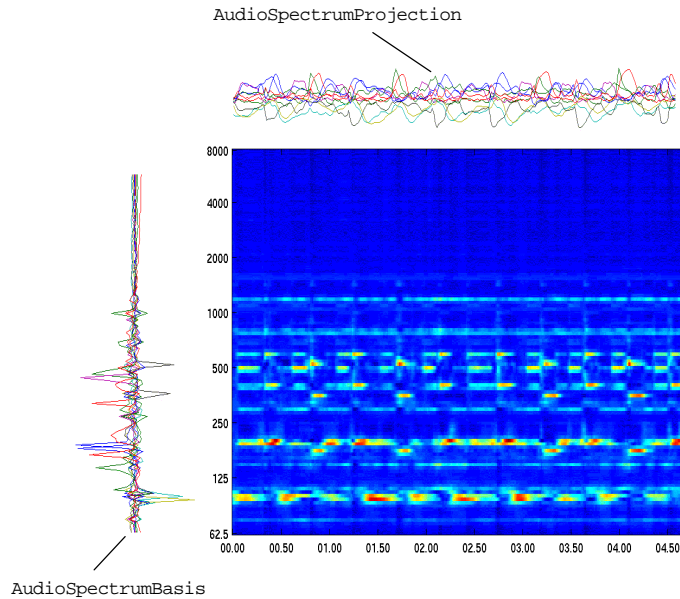
*Figure 28: A 10-basis component reconstruction showing most of the detail of the original spectrogram including guitar, bass guitar, hi-hat and organ notes. The left vectors are an* `AudioSpectrumBasis` *Descriptor and the top vectors are the corresponding* `AudioSpectrumProjection` *Descriptor. The required data storage is 10(M+N) values*

#### 3.3.1.9   Silence segment

The silence segment simply attaches the simple semantic of "silence" (i.e. no significant sound) to an Audio Segment. Although it is extremely simple, it is a very effective descriptor. It may be used to aid further segmentation of the audio stream, or as a hint not to process a segment.

### 3.3.2     High-level audio Description Tools (Ds and DSs)

Because there is a smaller set of audio features (as compared to visual features) that may canonically represent a sound without domain-specific knowledge, MPEG-7 Audio includes a set of specialized high-level tools that exchange some degree of generality for descriptive richness. The five sets of audio Description Tools that roughly correspond to application areas are integrated in the standard: audio signature, musical instrument timbre, melody description, general sound recognition and indexing, and spoken content. The latter two are excellent examples of how the Audio Framework and MDS Description Tools may be integrated to support other applications.

#### 3.3.2.1   Audio Signature Description Scheme

While low-level audio Descriptors in general can serve many conceivable applications, the spectral flatness Descriptor specifically supports the functionality of robust matching of audio signals. The Descriptor is statistically summarized in the AudioSignature Description Scheme as a condensed representation of an audio signal designed to provide a unique content identifier for the purpose of robust automatic identification of audio signals. Applications include audio fingerprinting, identification of audio based on a database of known works and, thus, locating metadata for legacy audio content without metadata annotation.

### 3.3.2.2  Musical Instrument Timbre Description Tools

Timbre Descriptors aim at describing perceptual features of instrument sounds. Timbre is currently defined in the literature as the perceptual features that make two sounds having the same pitch and loudness sound different.  The aim of the Timbre Description Tools is to describe these perceptual features with a reduced set of Descriptors. The Descriptors relate to notions such as "attack", "brightness" or "richness" of a sound.

Within four possible classes of musical instrument sounds, two classes are well detailed, and had been the subject of extensive development within MPEG-7. Harmonic, coherent, sustained sounds, and non-sustained, percussive sounds are represented in the standard. The HarmonicInstrumentTimbre Descriptor for sustained harmonic sounds combines the four harmonic timbral spectral Descriptors (see 3.3.1.7) with the LogAttackTime Descriptor. The PercussiveIinstrumentTimbre Descriptor combines the timbral temporal Descriptors (see 3.3.1.6) with a SpectralCentroid Descriptor. Comparisons between descriptions using either set of Descriptors are done with an experimentally-derived scaled distance metric.

### 3.3.2.3  Melody Description Tools

The melody Description Tools include a rich representation for monophonic melodic information to facilitate efficient, robust, and expressive melodic similarity matching. The Melody  Description Scheme includes a MelodyContour Description Scheme for extremely terse, efficient melody contour representation, and a MelodySequence Description Scheme for a more verbose, complete, expressive melody representation. Both tools support matching between melodies, and can support optional supporting information about the melody that may further aid content-based search, including query-by-humming.

The MelodyContour Description Scheme uses a 5-step contour (representing the interval difference between adjacent notes), in which intervals are quantized into large or small intervals, up, down, or the same. The Melody Contour DS also represents basic rhythmic information by storing the number of the nearest whole beat of each note, which can dramatically increase the accuracy of matches to a query.

For applications requiring greater descriptive precision or reconstruction of a given melody, the MelodySequence Description Scheme supports an expanded descriptor set and high precision of interval encoding. Rather than quantizing to one of five levels, the precise pitch interval (to cent or greater precision) between notes is kept. Precise rhythmic information is kept by encoding the logarithmic ratio of differences between the onsets of notes in a manner similar to the pitch interval. Arrayed about these core Descriptors are a series of optional support Descriptors such as lyrics, key, meter, and starting note, to be used as desired by an application.

### 3.3.2.4  General Sound Recognition and Indexing Description Tools

The general sound recognition and indexing Description Tools are a collection of tools for indexing and categorization of general sounds, with immediate application to sound effects. The tools enable automatic sound identification and indexing, and the specification of a Classification Scheme of sound classes and tools for specifying hierarchies of sound recognizers. Such recognizers may be used to automatically index and segment sound tracks. Thus the Description Tools address recognition and representation all the way from low-level signal-based analyses, through mid-level statistical models, to highly semantic labels for sound classes.

The recognition tools use the low-level spectral basis Descriptors as their foundation (see 3.3.1.8). These basis functions are then collected into a series of states that comprise a statistical model (the SoundModel Description Scheme), such as a hidden Markov or Gaussian mixture model. The SoundClassificationModel Description Scheme combines a set of SoundModels into a multi-way classifier for automatic labelling of audio segments using terms from a Classification Scheme. The resulting probabilistic classifiers may recognize broad sounds classes, such as speech and music, or they can be trained to identify narrower categories such as male, female, trumpet, or violin. Other applications include genre classification and voice recognition.

A SoundModelStatePath Descriptor consists of a sequence of indices to states generated by a SoundModel, given an audio segment. This simple Descriptor provides a compact description of a sound segment, and can be used for quick comparisons with other models. The SoundModelStateHistogram

Descriptor consists of a normalized histogram of the state sequence generated by a SoundModel. The descriptor may be used to compare sound segments via their histograms of their state activation patterns.

### 3.3.2.5 *Spoken Content Description Tools*

The spoken content Description Tools allow detailed description of words spoken within an audio stream. In recognition of the fact that current Automatic Speech Recognition (ASR) technologies have their limits, and that one will always encounter out-of-vocabulary utterances, the spoken content Description Tools sacrifice some compactness for robustness of search. To accomplish this, the tools represent the output and what might normally be seen as intermediate results of Automatic Speech Recognition (ASR). The tools can be used for two broad classes of retrieval scenario: indexing into and retrieval of an audio stream, and indexing of multimedia objects annotated with speech.

The Spoken Content Description Tools are divided into two broad functional units: the SpokenContentLattice Description Scheme, which represents the actual decoding produced by an ASR engine, and the SpokenContentHeader, which contains information about the speakers being recognized and the recognizer itself.

The SpokenContentHeader contains a number of components that are usable by any SpokenContentLattice. The header contains a WordLexicon Descriptor and a PhoneLexicon Descriptor, the former being an indexed list of tokens representing the repertoire of words of a recognizer, and the latter being the repertoire of phonetic components for a recognizer. There may be a ConfusionInfo Descriptor, which provides a confusion matrix and other insertion and deletion statistics for each entry in the PhoneLexicon. There must be a SpeakerInfo Descriptor, which conveys information about the person speaking in the audio, such as their vocabulary and phonetic repertoire, a characterization of their common speaking habits (ConfusionInfo), the language they're speaking, and information about them as a person (inherited from MDS's PersonType) such as their name. Additionally, within the SpokenContentHeader, there may be additional information on how the description was generated.

The SpokenContentLattice Description Scheme consists blocks of nodes. Nodes are connected by WordLinks or PhoneLinks, each of which links refers to a word or phone in the lexicon. Nodes are also indexed and are given a timeOffset from the beginning of the lattice. This highly flexible, but relatively compact, description format thus allows one to represent utterances with combinations of alternatives between words and phonemes. By combining these lattices, the problem of out-of-vocabulary words is greatly alleviated and retrieval may still be carried out when the original word recognition was in error. A simplified SpokenContentLattice is depicted in Figure 29.
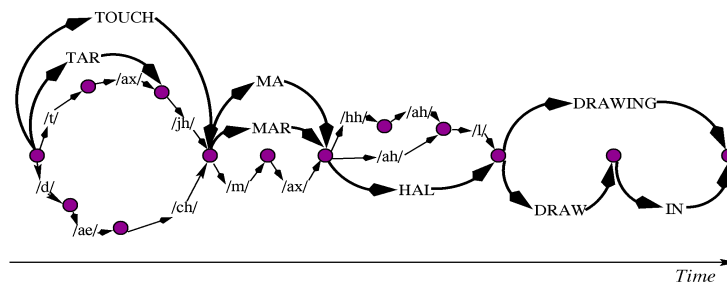


*Figure 29: A lattice structure for an hypothetical (combined phone and word) decoding of the expression "Taj Mahal drawing …". It is assumed that the name 'Taj Mahal' is out of the vocabulary of the ASR system*

Example applications for the spoken content Description Tools include:

- *Recall of audio/video data by memorable spoken events*. An example would be a film or video recording where a character or person spoke a particular word or sequence of words. The source media would be known, and the query would return a position in the media.

- *Spoken Document Retrieval.* In this case, there is a database consisting of separate spoken documents. The result of the query is the relevant documents, and optionally the position in those documents of the matched speech.

- *Annotated Media Retrieval.* This is similar to spoken document retrieval, but the spoken part of the media would generally be quite short (a few seconds). The result of the query is the media which is annotated with speech, and not the speech itself. An example is a photograph retrieved using a spoken annotation.

## 3.4 MPEG-7 Description Definition Language (DDL)

The main tools used to implement MPEG-7 descriptions are the Description Definition Language (DDL), Description Schemes (DSs), and Descriptors (Ds). Descriptors bind a feature to a set of values. Description Schemes are models of the multimedia objects and of the universes that they represent e.g. the data model of the description. They specify the types of the Descriptors that can be used in a given description, and the relationships between these Descriptors or between other Description Schemes

The DDL forms a core part of the MPEG-7 standard. It provides the solid descriptive foundation by which users can create their own Description Schemes and Descriptors. The DDL defines the syntactic rules to express and combine Description Schemes and Descriptors. According to the definition in the MPEG-7 Requirements Document the DDL is

> *'...a language that allows the creation of new Description SchemesDescription Schemes and, possibly, DescriptorsDescriptors. It also allows the extension and modification of existing Description SchemesDescription Schemes.'*

The DDL is not a modeling language such as Unified Modeling Language (UML) but a schema language to represent the results of modeling audiovisual data, i.e. DSs and Ds.

The DDL satisfies the MPEG-7 DDL requirements. It is able to express spatial, temporal, structural, and conceptual relationships between the elements of a DS, and between DSs. It provides a rich model for links and references between one or more descriptions and the data that it describes. In addition, it is platform and application independent and human- and machine-readable.

(Non-normative) DDL Parser applications will be required which are capable of validating description schemes (content and structure) and descriptor data types (both primitive (integer, text, date, time) and composite (histograms, enumerated types), against the DDL. The DDL Parsers must also be capable of validating MPEG-7 descriptions or instantiations, against their corresponding validated MPEG-7 schemas (DSs and Ds).

### 3.4.1 Context of development

The DDL design has been informed by numerous proposals and input documents submitted to the MPEG-7 DDL AHG since the MPEG-7 Call for Proposals in October 1998. It has also been heavily influenced by W3C's XML Schema Language and the Resource Description Framework (RDF).

At the 51$^{st}$ MPEG meeting in Noordwijkerhout in March 2000, it was decided to adopt XML Schema Language as the MPEG-7 DDL. However because XML Schema language has not been designed specifically for audiovisual content, certain extensions have been necessary in order to satisfy all of the MPEG-7 DDL requirements.

This overview presents the current status of the MPEG-7 Description Definition Language which is based on the XML Schema Recommendations published by the W3C in May 2001..

### 3.4.2 XML Schema Overview

The purpose of a schema is to define a class of XML documents by specifying particular constructs that constrain the structure and content of the documents. Possible constraints include: elements and their content, attributes and their values, cardinalities and datatypes. XML Schemas providea superset of the capabilities of DTDs.

The primary recommendation of the MPEG-7 community was that the DDL should be based on XML. Many solutions were available when the development started but none of them were considered

sufficiently stable  to provide a final solution. In April 2000, the W3C XML Schema Working Group published the Last Call Working Drafts of the XML Schema 1.0 specification. The improved stability of XML Schema Language, its potential wide-spread adoption, availability of tools and parsers and its ability to satisfy the majority of MPEG-7's requirements, led to the decision to adopt XML Schema as the basis for the DDL. However because XML Schema was not designed specifically for audiovisual content, certain specific MPEG-7 extensions were required. Consequently the DDL can be broken down into the following logical normative components:

- XML Schema Structural components;

- XML Schema Datatype components;

- MPEG-7 Extensions to XML Schema.

### 3.4.3    XML Schema: Structures

XML Schema: Structures is part 1 of the 2-part XML Schema specification. It provides facilities for describing the structure and constraining the content of XML 1.0 documents. An XML Schema consists of a set of structural schema components which can be divided into three groups. The primary components are:

- The Schema – the wrapper around the definitions and declarations;

- Simple type definitions;

- Complex type definitions;

- Attribute declarations;

- Element declarations.

The secondary components are:

- Attribute group definitions;

- Identity-constraint definitions;

- Named Group definitions;

- Notation declarations.

The third group is composed by the "helper" components which contribute to the other components and cannot stand alone:

- Substitution groups;

- Annotations;

- Wildcards.

Type definitions define internal schema components which can be used in other schema components such as element or attribute declarations or other type definitions. XML Schema provides two kinds of type definition component :

- simple types – which are simple data types (built-in or derived) which cannot have any child elements or attributes;
- complex types – which may carry attributes and have children elements or be derived from other simple or complex types.

Elements and attributes can then be declared which have are of types. New types can also be defined by derivation from existing types (built-ins or derived) through an extension or restriction of the base type.

Precise details of the use and application of these components can be found in the DDL Specification, ISO/IEC 15938-2  or the XML Schema: Structures Specification.

### 3.4.4    XML Schema: Datatypes

XML Schema:Datatypes is part 2 of the 2-part XML Schema specification. It proposes facilities for defining datatypes to be used to constrain the datatypes of elements and attributes within XML Schemas. It provides a higher degree of type checking than is available within XML 1.0 DTDs.

It provides:

- a set of built-in primitive datatypes;

- a set of built-in derived datatypes;

- mechanisms by which users can define their own derived datatypes.

A derived datatype can be defined from a primitive datatype or another derived datatype by adding constraining facets.

Precise details of the built-in datatypes and derivation mechanisms can be found in the DDL Specification, ISO/IEC 15938-2  or the XML Schema: Datatypes Specification.

### 3.4.5    MPEG-7 Extensions to XML Schema

The following features wereill need to be added to the XML Schema Language specification in order to satisfy specific MPEG-7 requirements :

- Array and amtrix datatypes – both fixed sibe and parametrized size;

- Built-in primitive time datatypes: basicTimePoint and basicDuration.

Precise details of the MPEG-7 extensions are available in the DDL Specification, ISO/IEC 15938-2.

MPEG-7-specific parsers have been developed by adding validation of these additional constructs to standard XML Schema parsers.

## 3.5 BiM (Binary Format for MPEG-7)

### 3.5.1    Introduction

XML has not been designed to deal ideally in a real-time, constrained and streamed environment like in the multimedia or mobile industry. As long as structured documents (HTML, for instance) were basically composed of only few embedded tags, the overhead induced by textual representation was not critical. MPEG-7 standardizes an XML language for audiovisual metadata. MPEG-7 uses XML to model this rich and structured data. To overcome the lack of efficiency of textual XML, MPEG-7 Systems defines a generic framework to facilitate the carriage and processing of MPEG-7 descriptions: BiM (Binary Format for MPEG-7). It enables the streaming and the compression of any XML documents.

### 3.5.2    BiM binary format is not dedicated to any specific XML language

BiM coders and decoders can deal with any XML language. Technically, the schema definition (DTD or XML Schema) of the XML document is processed and used to generate a binary format. This binary format has two main properties. First, due to the schema knowledge, structural redundancy (element name, attribute names, aso) is removed from the document. Therefore the document structure is highly compressed (98% in average). Second, elements and attributes values are encoded according to some dedicated codecs. A library of basic datatype codecs is provided by the specification (IEEE 754, UTF_8, compact integers, VLC integers, lists of values, aso...). Other codecs can easily be plugged using the type-codec mapping mechanism defined in the standard.

### 3.5.3    BiM is a schema oriented encoding scheme

One of the main technical advantages of the BiM binary encoding process is to be guided by schema information. In BiM, the schema is known both by the encoder and the decoder. This has several advantages compared to schema independent encoding scheme like WBXML or Zip or a specific encoding scheme.

First the binary format is deduced from the schema definition. There is no need to define coding tables or a specific encoding mechanism (like with the WBML encoding scheme). The binary format is inferred from the XML textual format. As a main result, very little extra work is needed to define the binary encoding of an XML language.

### 3.5.4 BiM is a pre-parsed format

In a typical Internet use case, the terminal has to validate a received document to recover default values, exact namespace information, aso… BiM performs the validation at encoder side and sends the document in pre-parsed format. Therefore, when encoded, a document has already reached a very good level of validity. Very few extra processing is needed on the receiver side to validate the received document.

### 3.5.5 BiM is a typed binary format

The validation process (performed by an XML validating parser) is used to associate a type information to every component of an XML document (attribute, element, leaf nodes), gives default values, aso. This mapping is performed at encoder side to improve compression ratio and to facilitate document processing. It is used to select the proper encoding scheme for each leaf of the XML document tree. The document values are therefore transmitted in a typed format and can directly be processed by the terminal without performing any string conversion (like the time consuming "atoi" function needed when working at textual level).

### 3.5.6 BiM is backward and forward compatible binary format

A BiM decoder can deal with evolution of XML languages. Technically, at encoding phase a level of compatibility is chosen for the bitstream. The encoding process adds necessary information to ensure that an old decoder will be able to skip unknown part of the bitstream. This feature allows also XML private extensions to be easily inserted within the original XML document without breaking interoperability. If forward compatibility is not needed, the redundancy is removed and the bitstream becomes more compact.

### 3.5.7 BiM allows a parameterized transmission of XML document.

Each document can be transmitted in one or more pieces. At the lowest level of granularity, each attribute value or document leaf can be modified to allow a minimal transmission in case of a minimal change in the sent document, like depicted in Figure 30.
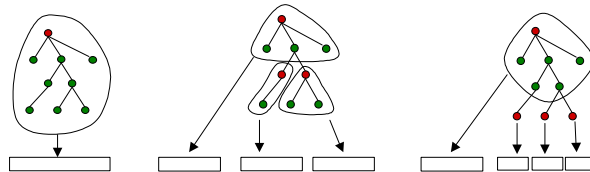


*Figure 30: Different streaming strategies of the same XML file*

For instance, the streaming capability of BiM enables to cut a large XML document in many pieces. These pieces can be delivered separately to the client. It is not required for the decoder to download (and keep in memory) the entire XML file before being capable of processing it. It can reduce both memory required at terminal side and consumed bandwidth. It improves overall quality of service and response time of XML based services.

In a streamed environment, a single XML document can be maintained at terminal side by constantly refreshing its sub parts. Different refresh rate can be imposed to different sub part of the same XML document. This updating capability allows to finely manage consumed bandwidth and to provide the best quality of service for the minimal bandwidth consumption.

### 3.5.8 BiM decoder can be efficiently processed at binary level

Unlike a zipped XML document, a BiM file can be processed directly at binary level. Moreover, document sub-trees can be skipped improving the overall performance of an application dealing with large documents. This optional "skipping" process can be triggered on the basis of element names, types or attribute values. This feature considerably improves browsing and searching through large XML files.



*Figure 31: Example of the fast access functionality*

### 3.5.9 BiM can be adapted to better suit a specific language

BiM is an open framework, which can receive, dedicated codec to better deal with specific domain constraints. For instance, BiM can receive quantification or zip compression algorithms.

## 3.6 MPEG-7 Terminal

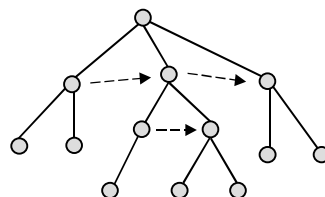### 3.6.1 Terminal architecture

MPEG-7 Systems provides the means to represent coded multimedia content descriptions. The entity that makes use of such coded representations of the multimedia content description is generically referred to as the "ISO/IEC 15938 terminal," "MPEG-7 terminal," or just "terminal" in short. This terminal may correspond to a standalone application or be part of an application system.

In Figure 32, there are three main architectural layers outlined: the application, the normative systems layer, and the delivery layer. MPEG-7 Systems is not concerned with any storage and/or transmission media (whose behaviours and characteristics are abstracted by the delivery layer) or the way the application processes the current description. The standard does make specific assumptions about the delivery layer, and those assumptions are outlined in 3.6.5.4. We describe the decoder architecture here in order to provide an overview.

### 3.6.2 General characteristics of the decoder

#### 3.6.2.1 General characteristics of description streams

An MPEG-7 terminal consumes description streams and outputs a – potentially dynamic – representation of the description called the current description tree. Description streams consist of a sequence of individually accessible portions of data named Access Units. An Access Unit (AU) is the smallest data entity to which "terminal-oriented" timing information (as opposed to "described-media oriented" timing information, the kind of time specified in the MDS) can be attributed. This timing information is called the "composition" time, meaning the point in time when the resulting current description tree corresponding to a specific Access Unit becomes known to the application. The timing information is carried by the delivery layer (see 3.6.5.4). The current description tree is schema-valid after processing each access unit.

A description consisting of textual Access Units is called a textual description stream and is processed by a textual decoder (see 3.6.2.2). A description stream consisting of binary Access Units is called a binary description stream and is processed by a binary decoder (see 3.6.2.3). A mixture of both formats in a single stream is not permitted. The choice of either binary or textual format for the description stream is application dependent. Any valid MPEG-7 description, with the exception of those listed in 3.6.6.4, may be conveyed in either format.

### 3.6.2.2  Principles of the textual decoder

The MPEG-7 Systems method for textual encoding, called TeM, enables the dynamic and/or progressive transmission of descriptions using only text. The original description, in the form of an XML document, is partitioned into fragments (see 3.6.5.1) that are wrapped in further XML code so that these resulting AUs can be individually transported (e.g., streamed, or sent progressively). The decoding process for these AUs does not require any schema knowledge. The resulting current description tree may be byte-equivalent to the original description if desired by the encoder, but it may also exhibit dynamic characteristics such that certain parts of the description are present at the decoder only at chosen times, are never present at all, or appear in a different part of the tree.

### 3.6.2.3  Principles of the binary decoder

Using the MPEG-7 Systems generic method for binary encoding, called BiM, a description (nominally in a textual XML form) can be compressed, partitioned, streamed, and reconstructed at terminal side. The reconstructed XML description will not be byte-equivalent to the original description. Namely, the binary encoding method does not preserve processing instructions, attribute order, comments, or non-significant whitespace. However, the encoding process ensures that XML element order is preserved.

The BiM, in order to gain its compression efficiency, relies on a schema analysis phase. During this phase, internal tables are computed to associate binary code to XML elements, types and attributes. This principle mandates the full knowledge of the same schema by the decoder and the encoder for maximum interoperability.

As with the textual decoder, the resulting current description tree may be topologically equivalent to the original description if desired by the encoder, but it may also exhibit dynamic characteristics such that certain parts of the description are present at the decoder only at chosen times, are never present at all, or appear in a different part of the tree.

### 3.6.3    Sequence of events during decoder initialisation

The decoder set-up is signalled by the initialisation extractor receiving a textual or binary DecoderInit. The signalling of the use of either binary or textual encoding is outside the scope of this specification, but a binary DecoderInit results in the expectation of binary Access Units, and a textual DecoderInit results in textual Access Units. The DecoderInit is passed to the systems layer by the delivery layer. The DecoderInit will typically be conveyed by a separate delivery channel compared to the description stream, which is also received from the delivery layer. The component parts of the description stream are discussed in the following section.
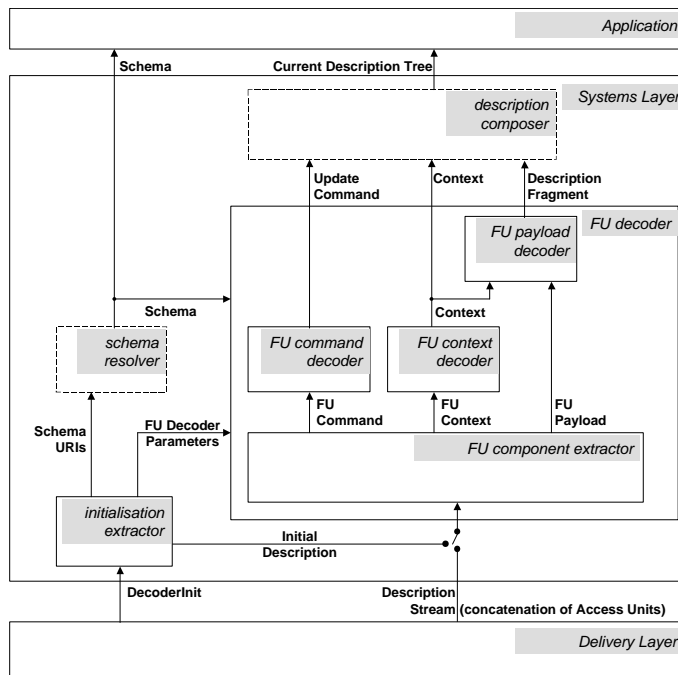
*Figure 32 Terminal Architecture.*
*Dashed boxes in the systems layer are non-normative. FU is an abbreviation for Fragment Update.*

The DecoderInit contains a list of URIs that identifies schemas, miscellaneous parameters to configure the decoder (FU Decoder Parameters, in Figure 32), and an Initial Description. There is only one DecoderInit per description stream. The list of URIs (Schema URIs, in Figure 32) is passed to a schema resolver that associates the URIs with schemas to be passed into the fragment update decoder. The schema resolver is non-normative and may, for example, retrieve schema documents from a network or refer to pre-stored schemas. The resulting schemas are used by the binary decoder and by any textual DDL parser that may be used for schema validation. If a given Schema URI is unknown to the schema resolver, the corresponding data types in a description stream are ignored.

The initial description has the same general syntax and semantics as an Access Unit, but with restrictions. The initial description initialises the current description tree without conveying it to the application. The current description tree is then updated by the Access Units that comprise the description stream. The initial description may be empty, since a schema-valid current description tree for consumption by the application need only be generated after the first Access Unit is decoded.

### 3.6.4   Decoder behaviour

The description stream is processed only after the decoder is initialised. An Access Unit is composed of any number of Fragment Update Units, each of which is extracted in sequence by the fragment update component extractor. Each Fragment Update Unit consists of:

- a Fragment Update Command that specifies the type of update to be executed (i.e., add, replace or delete content or a node, or reset the current description tree);

- a Fragment Update Context that identifies the data type in a given schema document, and points to the location in the current description tree where the Fragment Update Command applies; and

- a Fragment Update Payload conveying the coded description fragment to be added or replaced.

MPEG-7 Overview                                                                                                      51

A fragment update extractor splits the Fragment Update Units from the Access Units and emits the above component parts to the rest of the decoder. The fragment update command decoder generally consists of a simple table lookup for the update command to be passed on to the description composer. The decoded fragment update context information ('context' in Figure 32) is passed along to both the description composer and the fragment update payload decoder. The fragment update payload decoder embodies the BiM Payload decoder or, in the case of the TeM, a DDL parser, which decodes a Fragment Update Payload (aided by context information) to yield a description fragment (see Figure 32).

The corresponding update command and context are processed by the non-normative description composer, which either places the description fragment received from the fragment update payload decoder at the appropriate node of the current description tree at composition time, or sends a reconstruction event containing this information to the application. The actual reconstruction of the current description tree by the description composer is implementation-specific, i.e., the application may direct the description composer to prune or ignore unwanted elements as desired. There is no requirement on the format of this current description tree, e.g. it may remain a binary representation.

### 3.6.5 Issues in encoding descriptions

#### 3.6.5.1 Fragmenting descriptions

A description stream serves to convey a multimedia content description, as available from a (non-normative) sender or encoder, to the receiving terminal, possibly by incremental transmission in multiple Access Units. Any number of decompositions of the source description may be possible and it is out of scope of this specification to define such decompositions. Figure 33 illustrates an example of a description, consisting of a number of nodes, that is broken into two description fragments.

If multiple description fragments corresponding to a specific node of the description are sent (e.g., a node is replaced) then the previous data within the nodes of the description represented by that description fragment become unavailable to the terminal. Replacing a single node of the description effectively overwrites all children of that node.
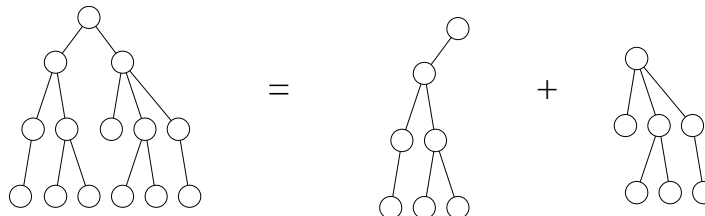


*Figure 33 Decomposition of a description into two description fragments*

#### 3.6.5.2 Deferred nodes and their use

With both the TeM and the BiM, there exists the possibility for the encoder to indicate that a node in the current description tree is "Deferred." A deferred node does not contain content, but does have a type associated with it. A deferred node is addressable on the current description tree (there is a Fragment Update Context that unambiguously points to it), but it will not be passed on to any further processing steps, such as a parser or an application. In other words, a deferred node is a placeholder that is rendered "invisible" to subsequent processing steps.

The typical use of deferred nodes by the encoder is to establish a desired tree topology without sending all nodes of the tree. Nodes to be sent later are marked as "deferred" and are therefore hidden from a parser. Hence, the current description tree minus any deferred nodes must be schema-valid at the end of each Access Unit. The deferred nodes may then be replaced in any subsequent Access Unit without changing the tree topology maintained internally in the decoder. However, there is no guarantee that a deferred node will ever be filled by a subsequent fragment update unit within the description stream.

### 3.6.5.3  Managing schema version compatibility with MPEG-7 Systems

It is very conceivable that a given schema will be updated during its lifetime. Therefore, MPEG-7 Systems provides, with some constraints, interoperability between different versions of MPEG-7 schema definitions, without the full knowledge of all schema versions being required.

Two different forms of compatibility between different versions of schema are distinguished. In both cases, it is assumed that the updated version of a schema imports the previous version of that schema. Backward compatibility means that a decoder aware of an updated version of a schema is able to decode a description conformant to a previous version of that schema. Forward compatibility means that a decoder only aware of a previous version of a schema is able to partially decode a description conformant to an updated version of that schema.

With both the textual and binary format, backward compatibility is provided by the unique reference of the used schema in the DecoderInit using its Schema URI as its namespace identifier.

When using the binary format, forward compatibility is ensured by a specific syntax. Its main principle is to use the namespace of the schema, i.e., the Schema URI, as a unique version identifier. The binary format allows one to keep parts of a description related to different schema in separate chunks of the binary description stream, so that parts related to unknown schema may be skipped by the decoder. In order for this approach to work, an updated schema should not be defined using the DDL "redefine" construct but should be defined in a new namespace. The Decoder Initialisation identifies schema versions with which compatibility is preserved by listing their Schema URIs. A decoder that knows at least one of the Schema URIs will be able to decode at least part of the binary description stream.

### 3.6.5.4  Reference consistency

The standard itself cannot guarantee reference (link) consistency in all cases. In particular, XPath-style references cannot be guaranteed to point to the correct node, especially when the topology of the tree changes in a dynamic or progressive transmission environment. With ID/IDRef, the system itself cannot guarantee that the ID element will be present, but during the validation phase, all such links are checked, and thus their presence falls under the directive that the current description tree must always be schema-valid. URI and HREF links are typically to external documents, and should be understood not to be under control by the referrer (and therefore not guaranteed).

### 3.6.6      Differences between the TeM and BiM

### 3.6.6.1  Introduction

BiM and TeM are two similar methods to fragment and convey descriptions as a description stream. While both methods allow one to convey arbitrary descriptions conformant to MPEG-7 MDS, Visual, and Audio, structural differences in the TeM- and BiM-encoded representation of the description as well as in the decoding process exist.

### 3.6.6.2  Use of schema knowledge

The TeM does not require schema knowledge to reconstitute descriptions; hence, the context information identifying the operand node on which the Fragment Update Command is applied is generated with reference to the current description tree as available to the decoder before processing the current fragment update. The TeM operates on an instantiation-based model: one begins with a blank slate (a single selector node) and adds instantiated nodes as they are presented to the terminal. Schema knowledge is, of course, necessary for schema validation to be performed.

The BiM relies upon schema knowledge, i.e., the FU decoder implicitly knows about the existence and position of all potential elements as defined by the schema, no matter whether the corresponding elements have actually been received in the instantiated description. This shared knowledge between encoder and decoder improves compression of the context information and makes the context information independent from the current description tree as available to the decoder. The BiM operates on a schema-based model: all possibilities defined by the schema can be unambiguously addressed using the context information, and as a payload is added, the instantiation of the addressed node is noted. The current description tree is built by the set of all of the instantiated nodes. One non-obvious consequence of this BiM model is that

numbering in the internal binary decoder model is "sticky": once an element is instantiated and thus assigned an address in the internal binary decoder model by its context, the address is unaffected by operations on any other nodes.

### 3.6.6.3 Update command semantics

The commands in TeM and BiM are named differently to reflect the fact that the commands operate on different models and have different semantics. The TeM commands have the suffix "node" because the TeM operates (nearly) directly on the Current Description Tree, and thus the removal of a node completely removes it from the tree. The BiM commands have the suffix "content" because the addressing on the Current Description Tree is by indirection, through an internal binary decoder model. Removal of an address, from the point of view of the application, via the current description tree, removes only the content (sub-elements and attributes), since the addressed node is still present in the internal model.

In the TeM, the commands are AddNode, ReplaceNode, and DeleteNode. The AddNode is effectively an "append" command, adding an element among the existing children of the target node. Insertion between two already-received, consecutive children of a node is not possible. One must replace a previously deferred node. By performing a DeleteNode on a node on the current description tree, the addressable indices of its siblings change appropriately.

In the BiM, the commands are AddContent, ReplaceContent and DeleteContent. The AddContent conveys the node data for a node whose path within the description tree is predetermined from the schema evaluation as described in the previous section. Hence, internally to the BiM decoder, the paths to (or addresses of) non-empty sibling nodes may be non-contiguous, e.g., the second and fourth occurrence of an element may be present. The "hole" in the numbering is not visible in the current description tree generated by the description composer. Hence, if the third occurrence of said element is added (using AddContent) in a subsequent access unit, it appears to any further processing steps as an "inserted" element in the current description tree, while it simply fills the existing "hole" with respect to the internal numbering of the BiM decoder. Similarly, DeleteContent does delete the node data, but does not change the context path to this node. ReplaceContent replaces node data and does not change the context path to this node either.

For both types of decoders, the "Reset" command reverts the description to the Initial Description in the DecoderInit.

### 3.6.6.4 Restrictions on descriptions that may be encoded

The TeM has limited capability to update mixed content models (defined in the DDL). Although it allows the replacement of the entire element, or the replacement of child elements, the mixed content itself cannot be addressed or modified.

Wildcards and mixed content models are not supported at all by the BiM. Therefore a schema that uses these mechanisms cannot be supported by the binary format.

### 3.6.6.5 Navigation

When navigating through a TeM description, at each step the different possible path is given by the element name, an index, and, possibly, a type identifier. The concatenation of that information is expressed (in a reduced form) by XPath.

In BiM, each step down the tree hierarchy is given by a tree branch code (TBC), whose binary coding is derived from the schema. The concatenation of all TBCs constitutes the context path information.

Both mechanisms in TeM and BiM allow for absolute and relative addressing of a node, starting either from the topmost node of the description or a context node known from the previous decoding steps.

### 3.6.6.6 Multiple payloads

With the BiM, for compression efficiency, there may be multiple payloads within a single Fragment Update Unit that implicitly operate on subsequent siblings to the operand node. This feature does not exist in the TeM.

### 3.6.7 Characteristics of the delivery layer

The delivery layer is an abstraction that includes functionalities for the synchronization, framing and multiplexing of description streams with other data streams. Description streams may be delivered independently or together with the described multimedia content. No specific delivery layer is specified or mandated by MPEG-7.

Provisions for two different modes of delivery are supported by this specification:

- Synchronous delivery – each Access Unit shall be associated with a unique time that indicates when the description fragment conveyed within this Access Unit becomes available to the terminal. This point in time is termed "composition time."

- Asynchronous delivery – the point in time when an Access Unit is conveyed to the terminal is not known to the producer of this description stream nor is it relevant for the usage of the reconstructed description. The composition time is understood to be "best effort," and the order of decoding AUs, if prescribed by the producer of the description, shall be preserved. Note, however, that this in no way precludes time related information ("described time") to be present within the multimedia content description.

A delivery layer (DL) suitable for conveying MPEG-7 description streams shall have the following properties:

- The DL shall provide a mechanism to communicate a description stream from its producer to the terminal.

- The DL shall provide a mechanism by which at least one entry point to the description stream can be identified. This may correspond to a special case of a random access point, typically at the beginning of the stream.

- For applications requiring random access to description streams, the DL shall provide a suitable random access mechanism.

- The DL shall provide delineation of the Access Units within the description stream, i.e., AU boundaries shall be preserved end-to-end.

- The DL shall preserve the order of Access Units on delivery to the terminal, if the producer of the description stream has established such an order.

- The DL shall provide either error-free Access Units to the terminal or an indication that an error occurred.

- The DL shall provide a means to deliver the DecoderInit information to the terminal before any Access Unit decoding occurs and signal the coding format (textual/binary) of said information.

- The DL shall provide signalling of the association of a description stream to one or more media streams.

- In synchronous delivery mode, the DL shall provide time stamping of Access Units, with the time stamps corresponding to the composition time (see section on synchronous delivery earlier in this sub-clause) of the respective Access Unit.

- If an application requires Access Units to be of equal or restricted lengths, it shall be the responsibility of the DL to provide that functionality transparently to the systems layer.

Companion requirements exist in order to establish the link between the multimedia content description and the described content itself. These requirements, however, may apply to the delivery layer of the description stream or to the delivery layer of the described content streams, depending on the application context:

- The DL for the description stream or the described content shall provide the mapping information between the content references within the description stream and the described streams.

- The DL for the description stream or the described content shall provide the mapping information between the described time and the time of the described content.

## 3.7 Reference Software: the eXperimentation Model

### 3.7.1 Objectives

The XM software is the frame work for the for all the reference code of the MPEG-7 standard. It implements the normative MPEG-7 components:

- Descriptors (Ds),
- Description Schemes (DSs),
- Coding Schemes (CSs),
- the Description Definition Language (DDL), and
- the BiM and TeM Systems components.

Besides the normative components, the simulation platform needs also some non-normative components, essentially to execute some procedural code to be executed on the normative data structures. The data structures and the procedural code together form the applications. For most Ds or DSs there is at least one application in the software framework, allowing the verification of the functionality of each normative component. The applications also show how to extract the metadata from the media content, or how the metadata can be used in simple application. Therefore, the XM implements only basic and elementary application types, and no real world applications. Furthermore, the XM software has only a command line interface, which does not allow any interaction during run-time.

The modules of the XM software are designed in a way, that all modules are using specified interfaces. This allows easy navigation through all the different modules for the various Ds and DSs, after understanding the structure one time. On the other hand, the usage of fixed interfaces allows to reuse and to combine individual modules in bigger application.

### 3.7.2 Extraction vs. Client Applications

Within the XM software framework, applications are related to one particular descriptor or description scheme. Because there are a lot of Descriptors and Description Schemes standardized, there are also a lot of applications integrated in the software framework. Applications, that are creating the descriptor (D) or description scheme (DS) they are testing, are called server or extraction applications. On the other hand, applications, which are using the D or DS under test (DUT), are called client applications. Extraction applications are needed if the D or DS is a low-level descriptor, which means that the description can be extracted from the multimedia content applying an automatic process. For high level Ds or DSs the extraction cannot be made in an automatic way. However, in most cases the extraction can be done based on preprocessed information. This means, that the extraction process reads this additional information besides the media data to populate the descriptions. Thus, the multimedia content set is extended by additional high level input data.

### 3.7.3 Modularity of the XM-software

By default the modules for all Ds and DSs are compiled to build one big executable which can then call the applications for an individual D or DS. However, the resulting executable becomes extremely big, because a lot of individual Ds and DSs are covered by the standard. Compiling the complete framework into one program results in an executable of more than 100 MBytes of size (in case debugging information is enabled). Therefore, the MPEG-7 XM software is designed in a way, that it supports partial compilation to allow to use only one single D or DS. On the other hand, in many cases it is desired to combine a subset of Ds or DSs. Furthermore, combining Ds and DSs is also required in case a DS is built in an hierarchical way from other Ds and DSs. In this scenario, it is not only important to allow partial compilation, but it is essential to design the software to allow as far as possible the reuse of code. As a conclusion, all applications are built from modules. These modules are:

- the media decoder class,

- the multimedia data class,

- the extraction tool class (only for extraction applications),

- the descriptor class,

- the Coding Scheme class, and

- the search tool class (only for client applications).

To increase the reusability, all this classes are using specified interfaces, which are independent from the D or DS they belong to. Thus, it should be possible to reuse, e.g., the extraction tool of a D or DS in another D or DS without knowing very deeply what is done in the included extraction tool. This is only possible if it is known how to use the interface of this extraction tool. The modules listed above are combined or connected to each other to form a processing chain. This is done in the application classes, which can be of the extraction (server) or client application type. In the following the modules are described.

### 3.7.4    Application modules

#### 3.7.4.1    Media decoders

The media decoder (MediaIO class) supports a wide range of possible input media formats. These are:

- audio data in WAV files,

- MPEG-1 video streams,

- motion vectors from MPEG-1 video streams (treated as still images),

- still images (JPEG, GIF, PNM, and many more),

- 4D key point lists (t,x,y,z),

- nD key point lists (t, x[0..n-1]), and

- other proprietary input formats for high level information

For this purpose the MediaIO class uses a set of external libraries which do not belong in all cases to the XM software source code tree. These libraries are the Afsp library for audio files, and ImageMagick for still images.

A special case are video sequences, because the decoded and uncompressed representation is to big to be held in memory. Therefore, the MediaIO class stores the decompressed images in temporary files, which can then be loaded using the routines for still images. The same mechanism is applied to motion vector information, but here the video sequence decoding is stopped after the motion vectors are available.

Because the MediaIO class is an interface to these libraries, the usage of the external libraries is not needed and not allowed in any other class of the XM software, enabling, e.g., that audio experts use the XM software without the video specific ImageMagick library.

#### 3.7.4.2    Multimedia data

The MultiMedia class holds the loaded media data in memory. Video sequences are, as described in section 3.7.2, not loaded into memory, but only the single frames of the sequence.

For still images the XM uses a reduced structure of the MoMuSys Vop data structure from the MPEG-4 Verification Model (VM). The data structure is defined in the file AddresLib/mymomusys.h. Key points are stored in a two dimensional linked list, one dimension for the time points (one frame) containing the second dimension, which includes all key points for this frame (see Media/MultiMedia.h). The Audio data structure is defined in the file Media/AudioFrame.h.

### 3.7.4.3 Extraction tool

The extraction tool performs the feature extraction for a single element of the multimedia database. The extraction process is a non-normative tool in the MPEG-7 standard. To perform the feature extraction, the extraction tool receives the references to the media data, which is the input for the extraction, and on the other hand the reference to the description, which stores the results from the extraction process.

Because in case of processing video sequences, it is not possible to provide all input data at the same time, the extraction is performed on a per frame basis. This means, that there are three function to be used for performing the extraction:

- InitExtracting  which is called before the first frame is processed,

- StartExtracting which is called in a loop over all frames to extract a part of the description, and

- PostExtracting which is called after all frames were processed. This is required if some part of the description can only be generated after all data was available (e.g., the number of frames in the sequence or frames).

The same interface is used in case audio data is processed. Here, the input data is more or less continuous (having only one sample at a time loaded has no meaning). Thus, the input is cut into time frames, which then can be processed one by one.

Besides the interfaces, the extraction classes have procedural code. In case of image or video extraction tools, the XM software uses the AddressLib which is a generic video processing library to perform the low level image processing tasks.

At the time being, the extraction tool is only be used in the extraction from media application type. As we will show later, it would also be possible to extract the D or DS under test from other description data. In this case, the extraction process could be performed with only one function call, i.e., without applying a loop iterating the input data for each time point or period.

### 3.7.4.4 Descriptor class

The descriptor classes hold the description data. In the XM software the classes for each D or DS represent directly the normative part of the standard. Besides providing the memory for the description also accessory function for the elements of the descriptions are available.

In the XM software there are two different ways of designing the D or DS class. In case of Visual Ds, this class uses a plain C++ class approach. In all other cases this class is implemented using a generic module, which is called the GenericDS in the XM software. This class is an interface from the C++ XM software to the instantiating DDL parser. Concrete, an XML parser providing the DOM-API (Data Object Model - Application Programming Interface) is used. Therefore, the GenericDS provides the interface from the XM to the DOM-API parser. The memory management for the description data is done by the DOM parser library. Both approaches can be combined using the functions ImportDDL and ExportDLL of the C++ implemented descriptor classes.

### 3.7.4.5 Coding Scheme

The Coding Scheme includes the normative encoder and decoder for a D or DS. In most cases the Coding Scheme is defined only by the DDL schema definition. Here, the coding is the dumping of the description to a file and the decoding is the loading and parsing of the description file into memory. The description is stored using the GenericDS class which is a wrapper to the DOM-API. Therefore, we can use the DOM-API parser library for encoding and decoding. Again, this functions are wrapped to the XM, using the GenericDSCS (CS = Coding Scheme) class. Besides the ASCII representation of the XML file, also a binary representation of descriptions (BiM: Binary format for MPEG-7) is standardized by MPEG-7. For delivery of MPEG-7 descriptions, the Systems part of the MPEG-7 standard specifies the BiM and the TeM (Textual format for MPEG-7)

Another approach is also be used in the Visual Group of MPEG-7. Here, each D also has an individual binary representation. This allows to specify the number of bits to be used for coding individual elements of the description. An example could be number of bits being used for coding each bin value of a histogram.

As the extraction tool, also the search tool represents a non-normative tool of the standard. It takes at the input one description from the database, and one description for the query, while the query does not need to be compliant to a normative MPEG-7 D or DS. The search tool navigates through the description and processes the required input date in a way that it is useful for the specific application.

Search tools are used in all client applications: search & retrieval, media transcoding, and description filter. In case of a search & retrieval application, the search tool compares the two input descriptions and computes a value for the distance between them. In the media transcoding application also media data are processed, i.e., the media information is modified basing on the description and the query. Because media data is processed, the search tool is called in the transcoding application in a media frame by media frame manner as it is done with the extraction tool.

### 3.7.5    Applications types in the XM software

*3.7.5.1  Extraction from Media*

In this section we describe the application types, which are implemented in the XM software.

The extraction from media application is of the extraction application types. Usually, all low level Ds or DSs should have an application class of this type. As shown in Figure 34 this application extracts the D/DS under test (DUT) from the media input data. First, the media file is loaded by the media decoder into the multimedia class, i.e., into the memory. In the next step, the description can be extracted from the multimedia class using the extraction tool. Then the description is passed through the encoder and the encoded data is written to a file. This process is repeated for all multimedia files in the media database.
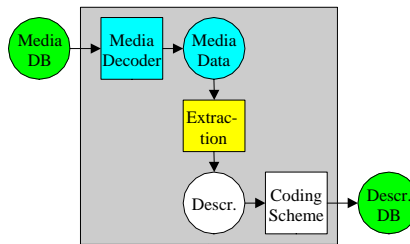


*Figure 34: Extraction from media application type. The description is extracted from the media input data.*

*3.7.5.2  Search & Retrieval Application*

The search & retrieval application, shown in Figure 35 is of the client application type. First all descriptions of the database, which might have been extracted using the extraction from media application, are decoded and loaded into the memory. Also the query description can be extracted from media using the extraction tool. On the other hand the query can also be loaded directly from a file. After having all input data, the query is processed on all elements of the database, and the resulting distance values are used to sort the data base with decreasing similarity to the query. Finally, the sorted list is written as a new media database to a file.
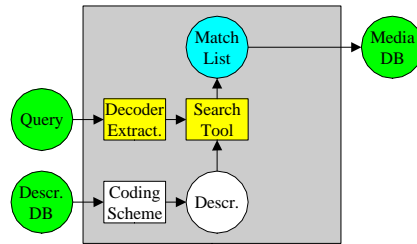
*Figure 35: Search and retrieval application type. A sorted media database is created from the descriptions with respect to a query (indexing).*

### 3.7.5.3 Media Transcoding Application

The media transcoding application is also of the client application type with consumes descriptive metadata. As shown in Figure 36, the media files and their description are loaded. Basing on the descriptions, the media data are modified (transcoded), and the new media database is written to a file. Furthermore, a query can be specified, which is processed on the description prior to the transcoding.
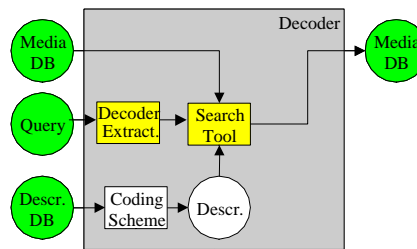


*Figure 36: Media transcoding application type. A transcoded media database is created from the original media database, the corresponding descriptions,and an optional query.*

### 3.7.5.4 Description Filtering Application

The description filtering application can either be of the extraction or client application type, depending whether the descriptor under test (DUT) is produced or consumed. In both cases the input description of the input database are filtered basing on the query. The resulting filtered descriptions are then written to the output files.
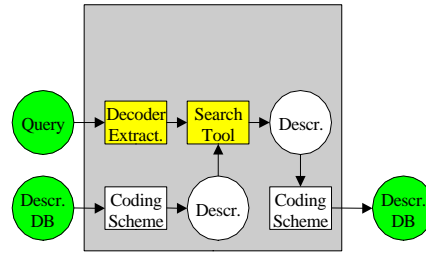
*Figure 37: Description filter application. A description is created, filtering the input description with respect to a query.*

### 3.7.6 The MPEG-7 key application model

#### 3.7.6.1 Definition of key applications

In the previous section the applications, which are implemented in the XM software, were described. These applications are also called key applications, because they are basic or elementary application types. They represent different application scenarios by implementing the key features of these application scenarios. In general, key applications are not necessarily real world applications because they only implement the representative and common task of the application scenarios. Details characterizing a specific real world application are not implemented.

Another important limitation of the XM software is the fact, that the XM software is a command line tool only, i.e., that the application, its inputs and outputs can only be specified when the XM is started. As a conclusion, the key applications do not support user interaction during run time.

#### 3.7.6.2 The interface model

After identifying the nature of key applications the second step is the design of an abstract key application model. Basing on the definition of key applications which was done using the interfaces, all possible inputs and outputs used by key applications can be collected. The resulting subset of the inputs and outputs is shown in Figure 38. Possible inputs are media databases, description databases, and queries. Possible outputs are media databases, and description databases. In the abstract model the semantics of the media database output is not distinguished, i.e, the list of best matching media files and the transcoded media database are not treated as individual output types, but they are in principle of the same kind.
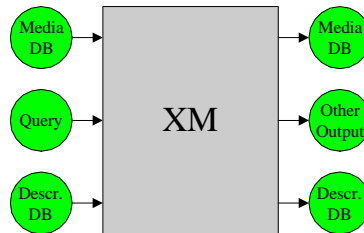


*Figure 38: Interface model for XM key applications. This model shows the superset of possible inputs and outputs of an XM key application.*

Besides the already used outputs, it is assumed that there will be also a corresponding output type for the query input. In Figure 38 this output has the name other output. Possible applications for this could be a refined query, e.g., for a browsing application. However, the usage of this output is still not clear and needs further investigations.

In principle, the key applications need to create only one of the output types. If an application produces two differnt output types, it can always be decomposed into separate applications and, therefore, it is not elementary. As a conclusion an application producing more then one output is not a key application.

In the following we use the interface model of the key applications for two purposes, which are the identification of new relevant key applications and the description of relations of key applications to real world applications.

### 3.7.7 Key applications vs. real world applications

As stated above, the key applications in the XM software are elementary application types. In general, combining the key applications will form complex applications. Because the key applications can have arbitrary combinations of inputs, the key application model is generic for this application area. Therefore, it is also possible that real world applications can be decomposed into processing networks consisting of the elementary key application blocks, and user interfaces providing user interaction and presentation of results.
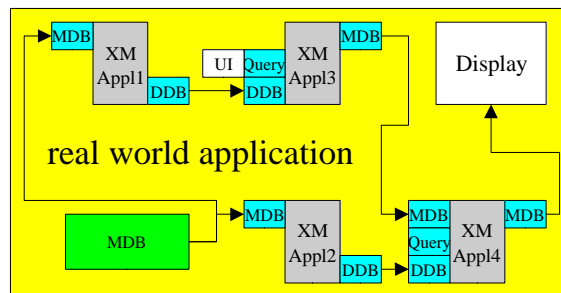


*Figure 39: Example of a real world application extracting two different descriptions (XM-Appl1, XM-Appl2). Basing on the first description the relevant content set is selected (XM-Appl3) which is then transcoded using the second description (XM-Appl4). (MDB = media database, DDB = description database).*

Figure 39 shows an example of a real world application. First, from a media database two features are extracted. Then, basing on the first feature, relevant media files are selected from the media database. The relevant media files are transcoded basing on the second extracted feature.

On the one hand, this is helpful for designing applications and products. On the other hand, also core experiments used in the standardization process are done basing on applications, which are in some cases more complex than key applications. In this cases, the decomposition or relation to the key applications can help do define evaluation criteria for the core experiments. The evaluation criteria are clear for the search & retrieval application, which is the retrieval rate, and for the extraction application, which is the computational complexity. In general also the bit stream complexity is an important evaluation criterion for all key applications. At the time being, the evaluation criteria are not clear for all possible key applications. However, answering this question for key applications seams more feasible than for proprietary applications.

## 3.8 MPEG-7 Conformance Testing

MPEG-7 Conformance Testing will include the guidelines and procedures for testing conformance of MPEG-7 implementations both for descriptions and terminals.

### 3.8.1 Conformance testing

#### 3.8.1.1 Conformance testing of descriptions

Figure 40 shows an overview of conformance testing of descriptions. The conformance testing consists of two stages: Systems testing and DDL testing. The Systems conformance testing involves decoding the description, which may be in binary or textual access unit form, and checking that the decoding produces a conformant textual XML description. In the case that the input description to the Systems processor is already in textual XML form, the System processor passes the description directly for DDL processing. The DDL conformance testing involves parsing the textual XML description and checking that the description is well-formed and valid according to the schema comprised from the MDS, Visual, Audio, Systems, and DDL parts of the standard.

The objective of the conformance testing of descriptions is to check the syntactic compliance with ISO/IEC IEC 15938 parts 1 – 5. As a result, the conformance testing of descriptions does not involve checking the semantics of the descriptions. For example, conformance testing of descriptions does not check whether the "name" field of a description actually contains the "true name" of an individual. However, the conformance testing of the description does determine whether the description is syntactically well-formed in the sense of XML processing and syntactically valid in the sense of conforming to the schema.
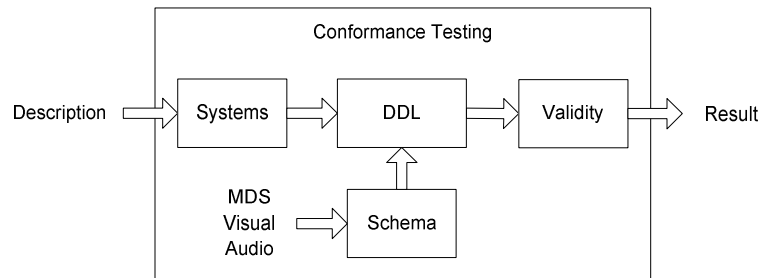


*Figure 40 - Overview of conformance testing of descriptions.*

#### 3.8.1.2 Conformance testing of terminals

Figure 41 shows an overview of conformance testing of terminals. The conformance testing involves the comparison of the result of processing a description using a reference terminal against the result using the test terminal. The reference terminal processes the description in terms of reference Systems and reference DDL processing. Likewise, the test terminal processes the description in terms of the test Systems and test DDL processing. The conformance testing of the terminal checks two things:

(1) Does the test terminal provide correct response for check of description validity, and

(2) Does the test terminal provide the same results for the reconstructed canonical XML (infoset) representation as the reference terminal.

In the case of an input description that is in the form of textual or binary access units, the Systems processing must first convert the description into a textual XML form. In the case of an input description that is already in textual XML form, the Systems processor passes the input description on for DDL processing. In either case, the textual XML form of the description is then operated on by DDL processor, which checks the description for well-formedness and validity. The DDL processesor takes as input the schema composed from the MDS, visual, audio, and other parts in order to allow the checking of the syntax of the textual XML description against the specifications of ISO/IEC 15938 Parts 1-5.
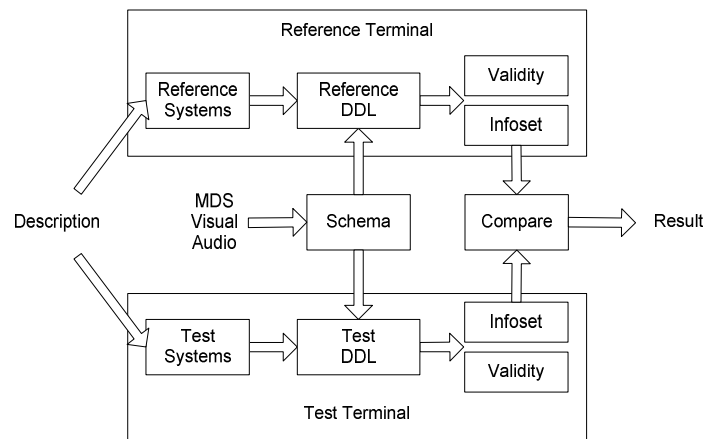
*Figure 41 - Overview of conformance testing of terminals.*

### 3.8.2    Interoperability points

Given the conformance testing procedures described above, the interoperability point in the standard corresponds to the reconstruction of a canonical XML representation of the description at the terminal. This allows for different possible implementations at the terminal in which different internal representations are used as long as the terminal is able to produce a conforming canonical XML representation of the description.

#### 3.8.2.1   Normative interfaces

##### 3.8.2.1.1   Description of the normative interfaces

The objective of this section is to describe MPEG-7 normative interfaces. MPEG-7 has two normative interfaces as depicted in Figure 42 and further described in this section.
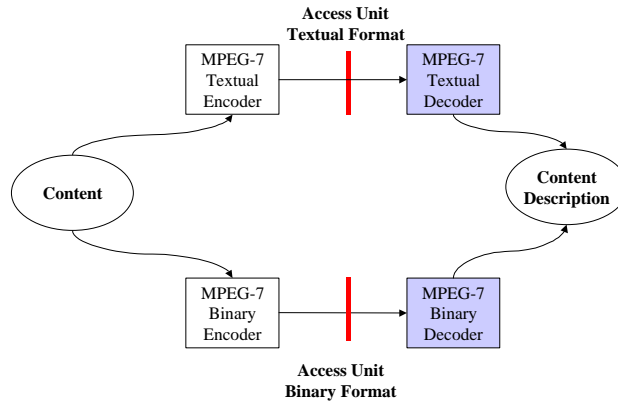
*Figure 42 - MPEG-7 Normative Interfaces*

**Content** : These are the data to be represented according to the format described by this specification. Content refer either to essence or to content description.

**MPEG-7 Binary/Textual Encoder** : These processes transform the content into a format compliant to this specification. The definition of these processes is outside the scope of this specification. They may include complex processing of the content such as features extraction.

**Textual Format interface** : This interface describes the format of the textual access units. The MPEG-7 Textual Decoder consumes a flow of such Access Units and reconstruct the content description in a normative way.

**Binary Format Interface** : This interface describes the format of the binary access units. The MPEG-7 Binary Decoder consumes a flow of such Access Units and reconstruct the content description in a normative way.

**MPEG-7 Binary/Textual Decoder** : These processes transform data compliant to this specification into a content description. The format of the reconstructed content description is outside the scope of this specification.

### 3.8.2.1.2    Validation of the standard

The objective of this section is to describe how proof can be established that thelossless  binary representation and the textual representation provide dual representations of the content. The process is described in Figure 43 and further described in this section.
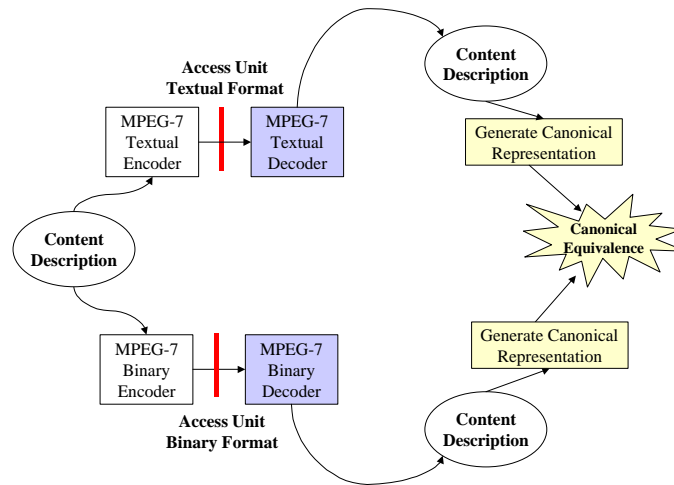
*Figure 43 - Validation process*

In addition to the elements described in section 3.8.2.1.1, the validation process involves the definition of a canonical representation of a content description. In the canonical space, content description can be compared. The validation process works as follows:

- A content description is encoded in a loss-less way in textual and in binary format, generating two different representation of the same entity.

- The two encoded descriptions are decoded with their respective binary and textual decoders.

- Two canonical descriptions are generated from the reconstructed content descriptions.

- The two canonical descriptions shall be equivalent.

- The definition of the canonical representation of an XML document is defined in Canonical XML[3].

## 3.9 MPEG-7 Extraction and Use of Descriptions

The MPEG-7 Extraction and Use of Descriptions Technical Report  gives examples of extraction and use of descriptions using Description Schemes, Descriptors, and datatypes as specified in ISO/IEC 15938. The following set of subclauses are provided for each description tool, where optional subclauses are indicated as (optional):

- Informative examples (optional): provides informative examples that illustrate the instantiation of the description tool in creating descriptions.

- Extraction (optional): provides informative examples that illustrate the extraction of descriptions from multimedia content.

- Use (optional): provides informative examples that illustrate the use of descriptions.

---

[3]  Canonical XML - Version 1.0, W3C Candidate Recommendation, 26 October 2000, http://www.w3.org/TR/xml-c14

ISO/IEC 15938-8 is meant to be a companion technical report for Part 5 (Multimedia Description Schemes) and Part 3 (Visual) of ISO/IEC 15938. As such, the content of the technical report is not easily understood without knowing about the corresponding technical specifications.

# 4. MPEG-7 Profiling

As happened for previous MPEG standards, profiling of the tools specified in the standard may play an important role in order the standard may be deployed with lower costs and complexity. This issue is currently under study; for the moment, only profiling of descriptions is being considered. Profiling the description consumption terminals is rather difficult due to the large range of possibilities that exists in terms of the way descriptions are consumed.

## 4.1 MPEG-7 Profiling

Relevant terms related to the specification of MPEG-7 profiling are:

- Profile

*The keyword to define profiles is 'functionality'; profiles are a set of tools providing a set of functionalities for a certain class of applications. A new profile should be defined if it provides a significantly different set of functionalities. It is not yet clear how many and which dimensions of profiling will be used for MPEG-7.*

The preliminary approach is that MPEG-7 profiles will be defined across MPEG-7 parts, at least for MDS, Audio and Visual.

- Level

*The keyword to define levels is 'complexity'; levels limit the complexity associated to a certain profile@level. A new level should be defined if it is associated to a significantly different implementation complexity.*

Possible dimensions of complexity are:

- Access units vs none
- BiM encoding vs none
- Linking (e.g. id/idref, headers) vs none
- Hierarchy vs graph relations
- Size of schema
- Size of descriptions
- Controlled vocabulary vs unconstrained
- Template descriptions vs unconstrained

Profiles should be named independently of applications to avoid sending wrong messages. Levels should be labeled using numbers.

## 4.2 Profiles under consideration

There are currently four MPEG-7 profiles under study; for more details, see the most recent version of the MPEG-7 Profiles under Consideration document.

### 4.2.1    Simple Profile

Generally, the application areas addressed by this Simple Profile are those where (currently) limited textual metadata (e.g., Title, Author, Copyright, Description/Abstract, Keywords, URL or Asset Identifier, etc.) are used to locate and subsequently access an entire multimedia asset or temporal segments thereof (aka "clips").  MPEG-7 provides additional textual metadata elements as well as

contextual structure to these elements supporting improved text-based search, browsing and filtering capabilities. Additionally, the profile supports links to digital rights management systems such as MPEG-21.

The requirements of the profile are to meet the needs of describing entire pieces (unsegmented) or temporal segments ("clips") of audiovisual material including audio, video, audiovisual, image, and multimedia content using text. Therefore, functionality is drawn only from ISO/IEC 15938 Part 5, Multimedia Description Schemes (MDS).

### 4.2.2 User Description Profile

The main, high level, functionality of this profile is description of users of multimedia content.

The description tools in this profile can be used to describe the personal preferences and usage patterns of users of multimedia content. Descriptions of users' preferences enable automatic discovery, selection and recommendation or recording of multimedia content. Preferences can be automatically inferred from the user's prior viewing and listening habits, which can in turn be derived from a usage history.

An important benefit of the tools in this profile is improved usability of a variety of multimedia devices through personalization: personalized multimedia services offered to the consumer, personalized multimedia content discovery, filtering and selection and personalized consumption of multimedia content. Such tools and services will help reduce the information overload that users may be faced with in the near future, due to the increasing availability of multimedia content from e.g. digital TV broadcast channels and the Internet.

Consumers of multimedia content will be able to capture their content preferences, likes and dislikes, and enjoy personalized television and music programming. Users may employ software agents to automatically figure out their personal tastes and automatically discover, select and recommend new multimedia content. A standardized format enables users to enter or update their preferences using one device, then import them into multiple other devices for instant customization. Users can carry a representation of their preferences in a secure smart card, or other type of removable storage. Furthermore, descriptions of users' preferences and usage history may be communicated to content or service providers, if the user allows this. In turn, content and service providers can introduce innovative content recommendation services that will enable them to attract and retain customers, by providing the right content. Device manufactures and middleware vendors may find new ways to differentiate their products, by providing improved content filtering and recommendation engines.

### 4.2.3 Summary Profile

Support for applications requiring a visual summary of content (e.g., an electronic program guide) d expand upon the MPEG-7 Simple Profile to include summarization and user description information.

### 4.2.4 Audiovisual Logging Profile

In this profile, the Simple Profile is extended with a subset of the MPEG-7 Audio and Visual tools to address the requirements of commercial audiovisual logging systems.

## 5. Current developments

Currently version 1 has been completed for parts 1 to 7. Part 8 (Extraction and use of descriptions) is still under development. Besides these, there are further work items under development (MPEG-7 version 2); these tools will be specified as amendments to the relevant parts of the standard, e.g. new audio tool in an amendment to the Audio part of the standard. The next subsections describe the different current developments for the different parts.

### 5.1 Systems

MPEG identified the need for extensions of the Systems part of the MPEG-7 standard and issued a Call for Proposals that was evaluatedin July 2002. The requirements for extensions consisted mainly in :

- Efficient representation of descriptions : whilst MPEG-7 Systems already provides a framework for efficiente representation of description, it is expected that this framework can be improved.

- Transmission mechanisms for MPEG-7 streams: MPEG-7 Systems shall support the transmission of MPEG-7 descriptions using a variety of transmission protocols.

The following high level issues have been raised during the evaluation of the MPEG-7 Systems extensions CfP:

- Optimized decoder management system: The contributions fit into the CfP framework. The granularity of the solution shall be assessed.

- Schema transmission: The overall schema management shall be considered and the proposed technical contributions need to be merged.

- DDL features: More DDL features shall be addressed (mixed content model, comments and entities).

- BiM and TeM commands: "insertion functionality" is required both in BiM and TeM for the cases where the description is not known beforehand by the encoder (dynamic descriptions). A copy command might be needed, but other commands can be emulated and their use has not been enough defined.

- Multiple Description Streams: The multiple stream features is required in the MPEG-7 context. The proposal showed a simple way of doing it but more complex scenarios were discussed. Several constraints shall be considered to preserve the coherency of the resulting description tree, notably with respect to synchronisation problems. These aspects have to be carefully considered in the near future. Moreover, some stream info is required but its components are not precisely defined and whether they should be standardized by the MPEG-7 Systems layer.

- Fragment references: Two contributions have been proposed, similar in the spirit. More evaluation is needed.

- Error resilience: It is not clear if handling error should be done at the delivery layer (even with slight modifications of the delivery layer requirements) or at the bistream level.

These issues are under study in several core experiments.

## 5.2 DDL

No DDL work is currently foreseen for version 2, although the Systems CfP evaluation (see above) may change the current situation.

## 5.3 Visual

Although ISO/IEC 15938-3 provides extensive functionality for describing visual features of multimedia content, new tools are needed to support additional features, such as color temperature and temporal variation of shape. It is also expected that tools significant improve performance over version 1 may be considered.

## 5.4 Audio

ISO 15938-4 Amd/1 PDAM specifies currently a modification of Spoken Content description tools and new description tools for audio signals quality.

### 5.4.1 Modification of Spoken Content description tools

Speech recognisers typically employ acoustic models and language models. The overall score of a particular speech recognition result is commonly calculated as the product of the likelihoods assigned by the acoustic models, the language model probability, a "fudge factor" to weight the importance of language model and acoustic model and a word insertion penalty that ensures that not only very short words are recognized.

Most ISO/IEC 15938-4 applications will only be interested in the overall probability assigned by the speech recognition engine. Some applications, however, will want to use application-specific language models or normalisation schemes. For these applications, a separate "acoustic only" score as produced by most speech recognition engines as intermediate result, is introduced.

### 5.4.2  Audio Signal Quality

Especially in the field of archiving and digital audio libraries it is necessary to know the quality of the audio signals that are in the archive in order to decide, if the material is suitable to broadcast or deliver. Furthermore, if multiple copies are available, retrieval of the best version can be guaranteed. Therefore, MPEG-7 Audio Amd. 1 specifies new tools to describe the quality of audio signals.

Several technical parameters to describe the quality can be extracted automatically, like balance, DC-Offset, or the bandwidth of the original recording. Some tools can be used to describe artefacts typical for audio signals coming from older recordings e.g. shellac or vinyl records. This information can be used later for automatic restoration.

### 5.4.3  Additional tools for MPEG-7 Audio version 2

There are additional tools under study to increase the current MPEG-7 Audio functionality, which may be developed to be part of Version 2 MPEG-7 Audio.

- Low Level Descriptor for tempo description

- Prosody for SpokenContent

- Syllables for Spoken Content

- Generic mechanism for compact / variable precision representation of Low Level data types (BAM)

## 5.5 MDS

ISO 15938-5 Amd/1 PDAM specifies currently some extensions to base types and new description tools for Linguist Description. Other tools and enhancements are under development.

### 5.5.1  New Base types

The `AudioDType` and `AudioDSType` are extended by providing each with an optional attributes that denotes which audio channels are used in computing the values of the `Audio D` and `Audio DS`, respectively. The convention of handling multi-channel signals (e.g. 5.1 surround format) is given in more details in ISO/IEC 15938-4:2001/PDAM 1 and Proposed ISO/IEC 15938-4:2001/Dcor1.

### 5.5.2  Linguistic Description tools

In order to create Linguist descriptions a new Multimedia Content Entity tool is specified to act as top-level element for accomodating the Linguistic description tools.

The Linguistic DS provides tools for encoding the semantic structure of linguistic data as part of or associated with multimedia/multimodal content, such as scenarios, transcriptions, critiques, and so forth. The DS describes the semantic structure of such linguistic data and their relation with the other data. Since such linguistic data exhibit the same range of structures as linguistic data in general, the Linguistic DS is designed to address the semantic structure of any kind of linguistic data.

A linguistic entity corresponds to a semantic structure in a logical form which may be regarded as a network (so called frame representation, typed feature structure, etc.). For instance, syntactic constituent "for a boy" has the following semantic structure:

[G,S]{:r:beneficiary(G,S) & :u:boy(S) & :u:singular(S)}

The above logical form may be regarded as a network as below:
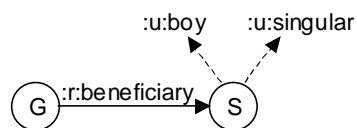
*Figure 44 Semantic structure of "for a boy"*

Thus an argument in the logical form is represented by a node in the network, a binary literal (an atomic formula with a binary relation) is represented by a solid arrow from its first argument to its second argument, and a unary literal (an atomic formula with a unary relation) is represented by a broken arrow from its argument to the predicate node.

Additional Classification Schemes are specified for use by Descriptors defined within the Linguistic DS.

### 5.6 Reference Software

Version 2 of the reference software will include the software corresponding to the tools defined in version 2 of the Systems, Audio, Visual and MDS standards.

### 5.7 Conformance Testing

Version 2 of Conformance Testing will include the conformance specification corresponding to the tools defined in version 2 of the Systems, Audio, Visual and MDS standards.

### 5.8 Extraction and Use of Descriptions

Version 2 of the Extraction and Use Technical Report will include extraction and use software corresponding to the tools defined in version 2 of the Audio, Visual and MDS standards.

## References

There are a number of documents available at the MPEG Home Page at http://mpeg.tilab.com/, including:

- Introduction to MPEG-7.

- MPEG-7 Requirements.

- MPEG-7 Applications.

- MPEG-7 Principal Concepts List.

- MPEG-7 CD, WD and XM documents: Systems, DDL, Visual, Audio and MMDS.

Information more focused to industry is also available at the MPEG-7 Alliance Web site at http://www.mpeg-industry.com.

Complete MPEG-7 schemas and description examples can be found at the MPEG-7 Schema page (http://pmedia.i2.ibm.com:8000/mpeg7/schema/).

MPEG-7 descriptions can be validated using the NIST MPEG-7 Validation Service (http://m7itb.nist.gov/M7Validation.html).

## Annexes

### Annex A - The MPEG-7 development process

The Moving Picture Coding Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio and their combination.

The purpose of MPEG is to produce standards. The first two standards produced by MPEG were:

- MPEG-1, a standard for storage and retrieval of moving pictures and audio on storage media (officially designated as ISO/IEC 11172, in 5 parts).

- MPEG-2, a standard for digital television (officially designated as ISO/IEC 13818, in 9 parts).

MPEG has recently finalized MPEG-4 Version 1, a standard for multimedia applications, that officially reached the status of International Standard in February 1999, with the ISO number 14496.

MPEG also started work on a new standard known as MPEG-7: a content representation standard for information search, scheduled for completion in Fall 2001. The Call for Proposals was issued in October 1998.

MPEG-1 has been a very successful standard. It is the de-facto form of storing moving pictures and audio on the World Wide Web and is used in millions of Video CDs. Digital Audio Broadcasting (DAB) is a new consumer market that makes use of MPEG-1 audio coding.

MPEG-2 has been the timely response for the satellite broadcasting and cable television industries in their transition from analogue to digital. Millions of set-top boxes incorporating MPEG-2 decoders have been sold in the last 3 years.

Recently standardized MPEG-4 provides (to authors) standardized tools for enabling the creation of content that has far greater reusability, (to network service providers) transparent information that can be interpreted and translated into the appropriate native signaling messages of each network with the help of relevant standards bodies, and (to users) higher levels of interaction with content, within the limits set by the author.

Since October 1996 MPEG is working on its fourth standard, called MPEG-7. MPEG considers of vital importance to define and maintain, without slippage, a work plan. This is the MPEG-7 work plan:

| Part | Title | WD | CD<br>PDAM | FCD<br>FPDAM | FDIS<br>FDAM | IS<br>AMD |
|------|-------|-----|------|-------|-------|-------|
| 1 | Systems | 12/99 | 10/00 | 02/01 | 07/01 | 09/01 |
| 1/Amd 1 | Systems Extensions | 07/02 | 12/02 | 03/03 | 07/03 | |
| 2 | DDL | 12/99 | 10/00 | 02/01 | 07/01 | 09/01 |
| 3 | Visual | 12/99 | 10/00 | 02/01 | 07/01 | 09/01 |
| 3/Amd 1 | Visual Extensions | 05/02 | 12/02 | 03/03 | 07/03 | |
| 4 | Audio | 12/99 | 10/00 | 02/01 | 07/01 | 09/01 |
| 4/Amd 1 | Audio Extensions | 12/01 | 05/02 | 10/02 | 03/03 | |
| 5 | Multimedia Description Schemes | 12/99 | 10/00 | 02/01 | 07/01 | 09/01 |
| 5/Amd 1 | Multimedia Description Schemes Extensions | 12/01 | 05/02 | 10/02 | 03/03 | |
| 6 | Reference Software | 12/99 | 10/00 | 02/01 | 07/01 | 09/01 |
| 6/Amd 1 | Reference Software Extensions | 12/01 | 05/02 | 10/02 | 03/03 | |

| 7 | Conformance Testing | 01/01 | 10/01 | 03/02 | 07/02 | 09/02 |
|---|---|---|---|---|---|---|
| 7/Amd 1 | Conformance Testing Extensions | | | | | |
| 8 | Extraction and use of descriptions (TR) | - | 07/01 | - | 07/02 | 09/02 |

Table A.1 - MPEG-7 work plan  (NB: The abbreviations are explained below)

The MPEG-7 call for proposals was issued in October 1998. This call, like all MPEG calls, was open to all interested parties, no matter whether they were within or outside of MPEG. It requested technology that proponents felt could be considered by MPEG for the purpose of the developing the MPEG-7 standard. The proposals were evaluated in an AHG Evaluation meeting in Lancaster 1999.

The proposals of technology received were assessed and, if found promising, incorporated in the so-called eXperimentation Models (XMs). A XM describes, in text and some sort of programming language, the operation of encoder and decoder. XMs are used to carry out simulations with the aim to optimize the performance of the Description Tools.

At the Maui meeting in December '99 MPEG reached sufficient confidence in the stability of the standard under development, and produced the some MPEG-7 Working Drafts (WDs).

The next steps until reaching the Final Drafts of International Standard (FDIS) starts with formal ballots by NBs that are usually accompanied by technical comments. These ballots are considered for the Commitee Drafts (CD) and Final Committee Drafts (FCD). This process entailed making changes. Finally, as Final Drafts of International Standard (FDIS), the standard is sent out for a final ballot, where NBs could only cast a yes/no ballot, without comments, within two months. After that, the FDIS became International Standard (IS) and is sent to the ISO Central Secretariat for publication.

## Annex B - Organization of work in MPEG

Established in 1988, MPEG has grown to form an unusually large committee. Some 300 to 400 experts take part in MPEG meetings, and the number of people working on MPEG-related matters without attending meetings is even larger.

The wide scope of technologies considered by MPEG and the large body of available expertise, require an appropriate organization. Currently MPEG has the following subgroups:

| Requirements | Develops requirements for the standards under development (currently, MPEG-4 and MPEG-7). |
|---|---|
| Delivery | Develops standards for interfaces between MPEG-4 applications and peers or broadcast media, for the purpose of managing transport resources. |
| Systems | Develops standards for the coding of the combination of individually coded audio, moving images and related information so that the combination can be used by any application. |
| Video | Develops standards for coded representation of moving pictures of natural origin. |
| Audio | Develops standards for coded representation of audio of natural origin. |
| SNHC (Synthetic-Natural Hybrid Coding) | Develops standards for the integrated coded representation of audio and moving pictures of natural and synthetic origin. SNHC concentrates on the coding of synthetic data. |
| Multimedia Description | Develops Structures for multimedia descriptions. This group only works for MPEG-7 and MPEG-21. |
| Test | Develops methods for and the execution of subjective evaluation tests of the quality of coded audio and moving pictures, both individually and combined, to test the quality of moving pictures and audio produced by MPEG standards. |

| | |
|---|---|
| Implementation | Evaluates coding techniques so as to provide guidelines to other groups upon realistic boundaries of implementation parameters. |
| Liaison | Handles relations with bodies external to MPEG. |
| HoD (Heads of Delegation) | The group, consisting of the heads of all national delegations, acts in advisory capacity on matters of general nature. |

Work for MPEG takes place in two different instances. A large part of the technical work is done at MPEG meetings, usually lasting one full week. Members electronically submit contributions to the MPEG FTP site (several hundreds of them at every meeting). Delegates are then able to come to meetings well prepared without having to spend precious meeting time to study other delegates' contributions.

The meeting is structured in 3 Plenary meetings (on Monday morning, on Wednesday morning and on Friday afternoon) and in parallel subgroup meetings.

About 100 output documents are produced at every meeting; these capture the agreements reached. Documents of particular importance are:

- Drafts of the different parts of the standard under development;

- New versions of the different Verification Models, that are used to develop the respective parts of the standard.

- "Resolutions", which document the outline of each agreement and make reference to the documents produced;

- "Ad-hoc groups", groups of delegates agreeing to work on specified issues, usually until the following meeting;

Output documents are also stored on the MPEG FTP site (drop.chips.ibm.com). Access to input and output documents is restricted to MPEG members. At each meeting, however, some output documents are released for public use, that can be accessed from the MPEG home page at http://mpeg.tilab.com.

Equally important is the work that is done by the ad-hoc groups in between two MPEG meetings. They work by e-mail under the guidance of a Chairman appointed at the Friday (closing) plenary meeting. In some exceptional cases, they may hold physical meetings. Ad-hoc groups produce recommendations that are reported at the first plenary of the MPEG week and function as valuable inputs for further deliberation during the meeting.

## Annex C - Glossary and Acronyms

| | |
|---|---|
| CD | Committee Draft |
| CE | Core Experiment |
| CS | Coding Scheme |
| D | Descriptor |
| DDL | Data Description Language |
| DS | Description Scheme |
| FAQ | Frequently Asked Question |
| FCD | Final Committee Draft |
| FDIS | Final Draft of International Standard |
| IS | International Standard |
| MMDS | Multimedia Description Schemes |
| MPEG | Moving Pictures Experts Group |

| NB | National Body |
|----|----|
| WD | Working Draft |
| XM | EXperimentation Model |

## Annex D - MPEG-7 FAQs

**Generic MPEG-7 FAQs**

*What is MPEG-7?*

MPEG-7 is formally called 'Multimedia Content Description Interface', a means of attaching metadata to multimedia content.

MPEG-7 will be a standardized description of various types of multimedia information. This description will be associated with the content itself, to allow fast and efficient searching for material that is of interest to the user. The standard does not comprise the (automatic) extraction of descriptions/features. Nor does it specify the search engine (or any other program) that can make use of the description.

In other words: MPEG-7 will specify a standard set of Descriptors that can be used to describe various types of multimedia information. MPEG-7 will also standardise ways to define other Descriptors as well as structures (Description Schemes) for the Descriptors and their relationships. This description (i.e. the combination of Descriptors and Description Schemes) shall be associated with the content itself, to allow fast and efficient searching for material of a user's interest. MPEG-7 will also standardise a language to specify Description Schemes, i.e. a Description Definition Language (DDL). AV material that has MPEG-7 data associated with it, can be indexed and searched for. This 'material' may include: still pictures, graphics, 3D models, audio, speech, video, and information about how these elements are combined in a multimedia presentation ('scenarios', composition information). Special cases of these general data types may include facial expressions and personal characteristics.

*Why is MPEG-7 needed?*

Nowadays, more and more audiovisual information is available, from many sources around the world. Also, there are people who want to use this audiovisual information for various purposes. However, before the information can be used, it must be located. At the same time, the increasing availability of potentially interesting material makes this search more difficult. This challenging situation led to the need of a solution to the problem of quickly and efficiently searching for various types of multimedia material interesting to the user. MPEG-7 wants to answer to this need, providing this solution.

The way MPEG-7 data will be used to answer user queries is outside the scope of the standard. In principle, any type of AV material may be retrieved by means of any type of query material. This means, for example, that video material may be queried using video, music, speech, etc. It is to the search engine to match the query data and the MPEG-7 AV description. A few query examples are:

- Music: Play a few notes on a keyboard and get in return a list of musical pieces containing (or close to) the required tune or images somehow matching the notes, e.g. in terms of emotions.

- Graphics: Draw a few lines on a screen and get in return a set of images containing similar graphics, logos, ideograms,...

- Image: Define objects, including colour patches or textures and get in return examples among which you select the interesting objects to compose your image.

- Movement: On a given set of objects, describe movements and relations between objects and get in return a list of animations fulfilling the described temporal and spatial relations.

- Scenario: On a given content, describe actions and get a list of scenarios where similar actions happen.

- Voice: Using an excerpt of Pavarotti's voice, and getting a list of Pavarotti's records, video clips where Pavarotti is singing or video clips where Pavarotti is present.

### *From whom or where did the demand for MPEG-7 come?*

The demand logically follows the increasing availability of digital audiovisual content. MPEG members recognized this demand, and initiated a new work item. There are many applications and application domains which will benefit from the MPEG-7 standard. A few application examples are:

- Digital libraries (image catalogue, musical dictionary,…)
- Multimedia directory services (e.g. yellow pages)
- Broadcast media selection (radio channel, TV channel,…)
- Multimedia editing (personalised electronic news service, media authoring)

The potential applications are spread over the following application domains:

- Education,
- Journalism (e.g. searching speeches of a certain politician using his name, his voice or his face),
- Tourist information,
- Cultural services (history museums, art galleries, etc.),
- Entertainment (e.g. searching a game, karaoke),
- Investigation services (human characteristics recognition, forensics),
- Geographical information systems,
- Remote sensing (cartography, ecology, natural resources management, etc.),
- Surveillance (traffic control, surface transportation, non-destructive testing in hostile environments, etc.),
- Bio-medical applications,
- Shopping (e.g. searching for clothes that you like),
- Architecture, real estate, and interior design,
- Social (e.g. dating services), and
- Film, Video and Radio archives.

### *What are the general applications for MPEG-7?*

It is often said that MPEG-7 will make the web more searchable for multimedia content than it is for text today. This would also apply to making large content archives accessible to the public (or to enable people to identify content to buy). The same information used for content retrieval may also be used by agents, for selection and filtering of broadcast or "push" material. Additionally, the meta-data may be used for more advanced access to the underlying data, by enabling automatic or semi-automatic multimedia presentation or editing.

### *What are the elements of MPEG-7?*

MPEG-7 main elements can be grouped in:

- Description Tools: Descriptors and Description Schemes for Audio (part 2), Visual (part 3) and Multimedia (part 4) content.

- Description Definition Language (part 2).

- System Tools (part 1).

- Reference Sofwtare –XM- (part 5).

### *What form does the "descriptions" of multimedia content in MPEG-7 take?*

The words 'descriptions' or 'features' represent a rich concept, that can be related to several levels of abstraction. Descriptions vary according to the types of data. Furthermore, different types of descriptions are necessary for different purposes of the categorization.

*Who participated in the development of the MPEG-7 standard?*

The people taking part in defining MPEG-7 represented broadcasters, equipment manufacturers, digital content creators and managers, transmission providers, publishers and intellectual property rights managers, as well as university researchers.

*What was the workplan for MPEG-7 (version 1)?*

The work plan was as follows:

| | |
|---|---|
| Working Draft | December 1999 |
| Committee Draft | October 2000 |
| Final Draft International Standard | July 2001 |
| International Standard | September 2001 |

*Does MPEG-7 include audio or video content recognition?*

The standardization of audiovisual content recognition tools is beyond the scope of MPEG-7. Following its principle 'specifying the minimum for maximum usability, MPEG-7 concentrated on standardizing a representation that can be used for description. Development of audiovisual content recognition tools will be a task for industries which will build and sell MPEG-7 enabled products.

In developing the standard, however, MPEG built some coding tools, just as it did with the predecessors of MPEG-7, namely MPEG-1, -2 and -4. Also for these standards, coding tools were built for research purposes, but they did not become part of the standard itself.

*Does MPEG-7 support audio or video content retrieval?*

In the same way that MPEG does not standardize the tools to generate the description, MPEG-7 does also not standardize the tools that use the description.

*Does the standard allow automatic extraction of descriptions as well as manual entry?*

The descriptions that conform to the MPEG-7 standard can be entered by hand, but they can also be automatically extracted. Some features can be best extracted automatically (color, texture), but for some other features ('this scene contains three shoes and that music was recorded in 1995') this is very hard or even impossible.

*What is the relationship between MPEG-7 and other MPEG activities?*

Where MPEG-1 and -2 concentrated almost entirely on compression, MPEG-4 moved to a higher level of abstraction in coding objects and using content-specific techniques for coding content. MPEG-7 moves to an even higher level of abstraction, a cognitive coding, some might say.

In principle, MPEG-1, -2, and -4 are designed to represent the information itself, while MPEG-7 is meant to represent information about the information (although there are areas common between MPEG-4 and -7 –see below). Another way of looking at it is that MPEG-1, -2, and -4 made content available. MPEG-7 allows you to find the content you need.

MPEG-7 can be used independently of the other MPEG standards - the description might even be attached to an analog movie. The representation that is defined within MPEG-4, i.e. the representation of audiovisual data in terms of objects, is however very well suited to what will be built on the MPEG-7 standard. This representation is basic to the process of categorization. In addition, MPEG-7 descriptions could be used to improve the functionalities of previous MPEG standards.

*So what happened to MPEG-5 and -6? (And how about 3?)*

MPEG-3 existed once upon a time, but its goal, enabling HDTV, could be accomplished using the tools of MPEG-2, and hence the work item was abandoned. So after 1,2 and 4, there was much speculation

about the next number. Should it be 5 (the next) or 8 (creating an obvious binary pattern)? MPEG, however, decided not to follow either logical expansion of the sequence, but chose the number of 7 instead. So MPEG-5 and MPEG-6 are, just like MPEG-3, not defined.

### *What are potential connections between MPEG-7 tools and existing MPEG tools?*

There are many possible connections between MPEG-4 tools and MPEG-7. Most of the content-specific tools contained in MPEG-4 have great potential because a model for the content is already specified: by choosing a method of coding, one selects the features that are important to the material. For example, if one encodes a sound by using sinusoidal tracks, then MPEG-7 asks which of those tracks are most significant in distinguishing the sound. It is a matter of abstraction up to the point of measuring similarity.

The Structured Audio tools also have a strong relationship to MPEG-7. They synthesize a sound from analready-existing description. The challenge in this case is to reach a suitable level of abstraction. There can be many, very different, descriptions which can be synthesized into perceptually indistinguishable sounds. It is clear that the models (e.g. of a musical instrument or an acoustic space) used directly within structured audio will not be sufficiently abstract or constrained.

### *When will MPEG-7 replace the existing MPEG-1, MPEG-2, and MPEG-4 standards?*

MPEG-7 will not replace MPEG-1, MPEG-2 or MPEG-4. It is intended to provide complementary functionality to these other MPEG standards: representing information about the content, not the content itself ("the bits about the bits") This functionality is the standardization of multimedia content descriptions.

### *If I want to get involved in MPEG-7, what do I need to know about the other MPEG standards?*

In principle, knowledge about the other three MPEG standards is not required for understanding or taking part in the MPEG-7 work. However, since some of MPEG-7's tools may be close to those of MPEG-4, some knowledge about them could be useful.

### *If I want to know more about the other MPEG standards, where do I look?*

You can start by taking a look at MPEG's home page (http://mpeg.tilab.com) which contains many useful references, including more lists with "Frequently Asked Questions" about MPEG activities.

### *Where can I find out more about MPEG-7?*

There are a number of documents available at the MPEG Home Page at http://mpeg.tilab.com, including

- MPEG-7 Requirements
- MPEG-7 Applications
- Introduction to MPEG-7
- MPEG-7 Evaluation results
- MPEG-7 Principal Concepts List
- MPEG-7 CD, WD and XM documents: Systems, DDL, Visual, Audio and MMDS parts.

There is also the MPEG-7 Alliance site, providing more industry oriented information. More information at http://www.mpeg-industry.com

### *What do the various acronyms used by MPEG-7 stand for?*

AHG – Ad Hoc Group

CE – Core Experiment

D – Descriptor

DDL – Description Definition Language

DS – Description Scheme

MDS – Multimedia Description Scheme

MPEG – Moving Picture Experts Group

MPEG-7 – MPEG standard for a "Multimedia Content Description Interface"

WD – Working Draft

XML – eXtensible Markup Language

XM – eXperimentatal Model

### If I want to know more about, be involved in, or give an input to the MPEG-7 (version 2) development process, whom should I contact?

You can contact any of the people listed below. To visit MPEG meetings you need to be on your national delegation, but the people listed below can explain how this works.

MPEG-7 Requirements: Fernando Pereira ([fp@lx.it.pt](fp@lx.it.pt))

MPEG-7 Systems: Olivier Avaro ([olivier.avaro@francetelecom.fr](olivier.avaro@francetelecom.fr))

MPEG-7 DDL: Jane Hunter ([jane@dstc.edu.au](jane@dstc.edu.au))

MPEG-7 Audio: Adam Lindsay ([atl@comp.lancs.ac.uk](atl@comp.lancs.ac.uk))

MPEG-7 Visual: Miroslaw Bober ([Miroslaw.Bober@vil.ite.mee.com](Miroslaw.Bober@vil.ite.mee.com))

MPEG-7 Multimedia DS: John R. Smith ([jsmith@us.ibm.com](jsmith@us.ibm.com))

MPEG-7 XM Software Implementation: Stephan Herrmann ([stephanh@lis.e-technik.tu-muenchen.de](stephanh@lis.e-technik.tu-muenchen.de))

MPEG-7 Alliance: Neil Day ([neil@mpeg-industry.com](neil@mpeg-industry.com))

### What does the Liaisons Subgroup do?

The Liaisons Subgroup maintains communications between MPEG and other standards bodies on topics of common interest.

### MPEG-7 Systems FAQs

There are currently no FAQs for MPEG-7 Systems. Questions should be forwarded to the general Systems reflector (gen-sys@advent.ee.columbia.edu).

### DDL FAQs

### What is the DDL?

The Description Definition Language (DDL) is the language which allows the creation of new Description Schemes (DSs) and Descriptors (Ds) and allows the extension and modification of existing DSs and Ds. The DDL defines the syntactic rules to express and combine Description Schemes and Descriptors. A DDL file is a schema which specifies the set of constraints that a valid MPEG-7 description must comply with. It is encoded in XML and more specifically uses the constructs defined in the W3C's XML Schema Language.

The DDL is specified in Part 2 of the MPEG-7 Specification.

*What is XML?*

Extensible Markup Language (XML) is a subset of SGML (ISO standard). Its goal is to enable generic SGML to be processed on the Web in the same way that is now possible with HTML. XML has been designed for ease of implementation compared to SGML.

XML defines document structure and embeds it directly within the document through the use of markups. A markup is composed of two kinds of tags which encapsulate data : open tags and closed tags. XML is similar to HTML but the tags can be defined by the user. The definition of valid document structure is expressed in a language called the DTD (Document Type Definition) or a Schema Language (e.g., XML Schema, RDF Schema).

The XML 1.0 specification defines the concepts of well formedness and validity. XML well-formedness requires that document tags are correctly nested. XML validity requires that a document follow the constraints expressed in its associated document type definition (DTD) or schema.

*What is XML Schema Language?*

XML Schema Language is a Recommendation developed by the XML Schema Working Group within the World Wide Web Consortium (W3C). It defines a language for defining the structure, content and semantics of XML documents. XML Schemas express shared vocabularies and allow machines to carry out rules made by people, expressed in the schema language. An XML Schema is itself an XML document. Details of XML Schema are available from W3C at: http://www.w3.org/XML/Schema.

*What is the relevance of XML Schema language to the DDL?*

The main recommendation of the MPEG-7 community was to use a schema language based on XML for the DDL. Many solutions were available when the development started but none of them were considered stable enough or complete enough (DSD, SOX, DDML, DTD, RDF). Initially the DDL group decided to develop its own language whilst keeping track of W3C's XML Schema developments. However the improved stability of XML Schema Language, its potential wide-spread adoption, availability of tools and parsers and its ability to satisfy the majority of MPEG-7's requirements, led to the decision to adopt XML Schema as the basis for the DDL. At the 51st MPEG meeting in Noordwijkerhout in March 2000, it was decided to adopt XML Schema Language [4,5,6] as the MPEG-7 DDL. However because XML Schema language has not been designed specifically for audiovisual content, certain extensions are necessary in order to satisfy all of the MPEG-7 DDL requirements. Hence the MPEG-7 DDL adopted most the of the XML Schema specification and added MPEG-7-specific mechanisms on top of it.

*What are the MPEG-7 specific extensions added to XML Schema in the DDL?*

The features which have been added to the XML Schema Language specification in order to satisfy specific MPEG-7 requirements are:

- Array and Matrix datatypes;
- Built-in datatypes for basicTimePoint and basicDuration.

For more details of these, you will need to read the MPEG-7 DDL IS - ISO/IEC 15938-2/IS.

*What is the current status of XML Schema?*

TheXML Schema specifications are W3C Recommendations. Details of the W3C document process can be found at http://www.w3.org/Consortium/Process-20010208/. Recommendations are the final stage in W3C's documentation process and indicate that there is a consensus within the W3C that the specification is appropriate for widespread deployment.

*What is the relationship between the DDL Group and the XML Schema Working Group?*

There is a liaison between these two groups who facilitates communication between them. The MPEG-7 community provides implementation feedback and liaises with the W3C and the XML Schema Working

Group to ensure that MPEG-7 requirements are satisfied as far as possible. The MPEG-7 DDL will reflect any changes made to XML Schema as new versions are released.

*How is the DDL used within MPEG-7?*

The DDL is used by the MDS, Video and Audio groups to define Description Schemes (DSs) and Descriptors (Ds). In particular it defines structural and cardinality constraints on DSs and datatyping constraints on Ds.

*What kind of constraints can I express with the DDL ?*

With the DDL you can express structural, cardinality and data type constraints. Structural constraints specify the rules that a valid description should respect in terms of inclusion of child elements in parent elements or the association of attributes with elements. Cardinality constraints specify the number of times an element may occur. Data type constraints specify the type and the possible values for data or Descriptors within the description.

*What are the main features of the DDL?*

The DDL allows you to define two kinds of type definitions:

- simple types – which are simple data types (built-in or derived) which cannot have any child elements or attributes;
- complex types – which may carry attributes and have children elements or be derived from other simple or complex types.

These define internal schema components which can be used in other schema components such as element or attribute declarations or other type definitions.

Moreover the DDL allows you to reuse existing complexTypes or simpleTypes in a way similar to inheritance in object oriented programming. New types can be defined by derivation from existing types (built-ins or derived) through extension or restriction of the base type. Elements and attributes can then be declared which have these new types.

*What is the purpose of a DDL parser ?*

The parser checks the validity of a description according to the rules defined in the DDL Schema to which the description must conform. The parser is first initialized with a schema (a DDL file). It then "parses" a description and reports any errors (both with respect to well-formedness, as well as non-conformance to structural and datatyping constraints).

*Is there an MPEG-7 parser available ?*

At the moment there is an MPEG7-specific parser under development by Expway. This parser has been developed by adding the validation of MPEG-7 extensions to the Xerces open source XML Schema parser. There are also a number of XML Schema Language parsers available which are capable of validating MPEG-7 descriptions except when they contain MPEG-7-explicit extensions:

- XSV - Open Source Edinburgh Schema Validator (written in Python);
- XML Spy - Validating XML Editor;
- Xerces - Open source XML Parsers in Java and C++.

*Where do I go if I have more questions about the DDL?*

You can have a look at the MPEG-7 DDL Specification which is Part 2 of the MPEG-7 specification: ISO/IEC 15938-2.How do I join the MPEG-7 DDL AHG?

If you are a member of MPEG, you may join the DDL Group by participating its meetings which are held at the MPEG meetings. You may also by join the DDL Ad Hoc Group, by subscribing to its reflector. To

do this send an email to [majordomo@darmstadt.gmd.de](mailto:majordomo@darmstadt.gmd.de) with the only message being "subscribe mpeg7-ddl". If you are not yet a member of MPEG, please see find information about joining at [http://mpeg.tilab.com](http://mpeg.tilab.com).

**MPEG-7 Visual FAQs**

*What do the MPEG-7 Visual Descriptors describe?*

The MPEG-7 Visual Descriptors describe the visual features of images and video such as color, texture, shape and motion. etc.

*What kinds of visual descriptors are specified in MPEG-7 Visual Descriptors standard?*

The MPEG-7 Visual Descriptors include the Dominant Color descriptors, Color Space Descriptor, Region based Descriptor, Homogeneoud Texture Descriptor, ….

*How are the MPEG-7 Visual Descriptors defined?*

An MPEG-7 Visual Descriptor defines the syntax and the semantics of a feature representation.

*What functionalities are provided by the MPEG-7 Visual Descriptors?*

The MPEG-7 Visual Descriptors provides content-based image or video retrieval.

*How are the MPEG-7 Visual Descriptors computed or instantiated?*

An MPEG-7 Visual Descriptor values are computed via a feature extraction process and represented in the standardized forms, which are binary data.

*What are the MPEG-7 Visual Descriptors related with the MPEG-7 Multimedia Description Schemes (MDS)?*

An MPEG-7 Visual Descriptor is the component of an MDS which specifies the structure and semantics of the relationships between its components, which may be both Descriptors and Description Schemes.

*How can the MPEG-7 Visual Descriptors be used in applications?*

The MPEG-7 Visual Descriptors can be used in pull applications such as image and video retrieval. In addition, the push applications are also the areas where the MPEG-7 Visual Descriptors are used for content filtering.

*Why do the visual descriptors need be standardized?*

The standardized description of visual features ensures that the interoperability is guaranteed in delivering and exchanging the visual description data between different venders.

*Which subgroup in MPEG is responsible for standardizing the MPEG-7 Visual Descriptors?*

The standardization activities on MPEG-7 Visual Descriptors are taken place in the MPEG Video subgroup.

*Does a description of the feature extraction algorithm for the MPEG-7 Visual Descriptors exist? Is it available?*

The detailed descriptions of teature extraction algorithms are described in the *MPEG-7 part 8: Extraction and use of descriptions* Technical Report.

*How can I make technical proposals with my visual descriptors for MPEG-7 Visual Descriptors?*

A new visual descriptors can be submitted to MPEG meetings and the newly proposed descriptor is evaluated according to the MPEG standardization procedure in the MPEG-7 Standardization schedule.

*What are the Core Experiments in the MPEG-7 Visual Descriptors?*

The Core Experiments are set up in order to investigate proposed visual descriptors. The results of the Corement Experiments are reviewed by the MPEG Video subgroup every MPEG meeting.

*Where can I find the software for the MPEG-7 Visual Descriptors?*

Integrated within the XM software (see XM software implementation FAQs below)

**MPEG-7 Audio FAQs**

*What are specific functionalities for MPEG-7 audio? (or, what can I tell my organization MPEG-7 is good for?*

Indexing music, sound effects, and spoken-word content in the audio-only arena. MPEG-7 enables query-by-example such as query-by-humming. In addition, audio tools play a large role in typical audio-visual content in terms of indexing film soundtracks and the like. If someone wants to manage a large amount of audio content, whether selling it, managing it internally, or making it openly available to the world, MPEG-7 is potentially the solution.

For more details, please see the MPEG-7 Applications Document for a non-exhaustive list of representative examples. We are actively seeking input in this area, so if you are considering a potential MPEG-7 application that is not in the document, please contact the editor.

*How do I join the MPEG-7 Audio Issues AHG?*

Send a message with "subscribe" in the Subject line to: mpeg-7-aud-request@meta-labs.com

*Where do I go if I have more questions?*

If you have any questions, please do not hesitate to join the MPEG-7 Audio Issues AHG and direct your questions to the reflector. Chances are, there are others who share your concerns, and the AHG exists specifically to identify those concerns. If you must reach an individual, Adam Lindsay <atl@comp.lancs.ac.uk> will be happy to answer any questions regarding MPEG-7.

**MDS FAQs**

*What is the MPEG Multimedia Description Scheme (MDS) Group and why was it established?*

The Multimedia Description Schemes (MDS) Group is a working group of MPEG formally established in July, 1999 to develop Multimedia Description Schemes as part of the MPEG-7 "Multimedia Content Description Interface" standard. Currently it is also working in MPEG-21 besides in MPEG-7 version 2.

*What are Multimedia Description Schemes?*

Multimedia Description Schemes refer to specific metadata structures being developed for MPEG-7 that are used to describe and annotate audio-visual data. In the MPEG-7 standard, the Multimedia Description Schemes are categorized as pertaining to visual, audio, or generic description. In general, the Multimedia Description Schemes (DSs) can contain MPEG-7 Descriptors and other DSs and can be extended for domain-specific applications.

*What do the Multimedia Description Schemes describe?*

Multimedia Description Schemes provide a way to describe in XML the important concepts related to audio-visual data in order to facilitate the searching, indexing and filtering of audio-visual data. Many of the audio-visual concepts are defined in the MPEG-7 Principal Concepts List. The Description Schemes (DSs) are designed to describe both the generic aspects of audio-visual data management and the specific content and features of the audio-visual material. The generic DSs provide a way to specify immutable meta-data related to the creation, production, usage and management of the audio-visual data. For example, these DSs can be used to describe the title and author of an audio-visual program. The audio-visual content-specific DSs provide a way to specify the content directly at a number of levels including signal structure, features, models and semantics. Other audio-visual specific DSs are designed to allow efficient navigation and access of the audio-visual data.

Some examples of the different types of Multimedia Description Schemes are given as follows:

- Semantics – the semantics-based DSs provide a way to describe what is depicted in the audio visual data from the real world, such as objects, actions, and events.

- Structure – the structure-based DSs provide a way to describe the structure of the audio-visual data, such as the regions, segments, frames, shots, and spatial and temporal relationships. Some of the structure-based DSs contain MPEG-7 Descriptors that describe perceptual features of the audio-visual data, such as color, texture, shape, motion, pitch and rhythm.

- Models – the model-based DSs describe feature classes and their relationships with semantic information. The model-based DSs also form classifiers and models that relate the structural and semantic aspects of the audio-visual content.

- Navigation and Access – Some of the DSs are designed specifically to address navigation, access and browsing by providing structured summaries and management of different variations of the audio-visual programs.

*How are the Multimedia Description Schemes defined?*

Description Schemes (DSs) are defined using the MPEG Description Definition Language (DDL). DDL allows the creation and extension of DSs. DDL is a language similar to XML-schema, but supports additional features such as polymorphism, data value inheritance based on spatio-temporal inclusion interval, and extended spatio-temporal and audio-visual data types.

*What functionalities are provided by Multimedia Description Schemes?*

Multimedia Description Schemes provide a standard set of data types and description structures needed for generating XML descriptions of audio-visual content. The descriptions are human-readable and can be searched, transmitted and filtered in applications that deal with audio-visual data.

*How are the MPEG-7 Multimedia Description Schemes computed or instantiated?*

Many Description Schemes (DSs) are computed automatically by processing the audio-visual data directly, such as by using feature extraction algorithms or content analysis methods. Other descriptions are generated manually by human annotation or by processing external meta-data related to the audio-visual material.

*What is the data model behind the Multimedia Description Schemes?*

The full set of Multimedia Description Schemes, which are related to each other through linking, form a data model or schema for audio-visual content description. At both the conceptual- and implementation-levels, the data model is object-oriented and employs modeling principals such as generalization, association, aggregation and spatio-temporal composition. The Description Schemes (DSs) ultimately generate description data that is in XML format.

*I am familiar with other multimedia meta-data models and standards. What is new with     MPEG-7 Multimedia Description Schemes?*

MPEG-7 Description Schemes (DSs) are defined using the MPEG Description Definition Language (DDL).  Multimedia Description Schemes are unique in their modeling of audio-visual content directly at a number of levels including audio-visual signal structure, features, models and semantics.

*How can the Multimedia Description Schemes be used in applications?*

Description Schemes (DSs) can be used for searching, filtering and browsing audio-visual material. Since they provide a more complex data model for content description than Descriptors, they are particularly useful in producing rich descriptions of the audio-visual content that involve a number of Descriptors and DSs.

*How can the MPEG-7 Multimedia Description Schemes be extended for my application?*

The MPEG Description Definition Language (DDL) can be used to define, extend and generate new Description Schemes (DSs).  By using the object-oriented aspects of the DS model, it is possible to extend the DSs to develop customized, application specific DSs that are outside of the MPEG-7 standard.

*How are the MPEG-7 Multimedia Description Schemes different from MPEG-7 Descriptors?*

Although there is no formally stated distinction, generally, MPEG-7 Descriptors address the low-level perceptual features of audio-visual data and are meant to be extracted automatically.  In many cases Descriptors correspond to the perceptual attributes of audio-visual data and its regions and segments, such as to describe the color of a region.

Furthermore, the Descriptor data is typically not human-readable since the Descriptor provides a compact binary description of a feature of the audio-visual data.  On the other hand, the Description Schemes (DSs) are defined using DDL and produce human-readable XML descriptions.

*Who are the members of the MPEG Multimedia Description Scheme (MDS) Group?*

The Multimedia Description Schemes (MDS) Group is made up of about 40-50 people that attend the MPEG meetings and >200 people that participate in the various MDS Group Ad Hoc Group mail reflectors. The members represent a significant diversity of interests including content providers, software companies, hardware manufacturers, research organizations, and academic institutions.

*How many Multimedia Description Schemes are standardized by MPEG?*

At present, approximately more than one hundred Description Schemes (DSs) have been standardized. These DSs relate to over 130 important concepts from the audio-visual domain as identified in the MPEG-7 Principal Concepts List.

*What is the process for standardization Multimedia Description Schemes in MPEG?*

The standardization process requires that an initial proposal be made to MPEG for the specific Description Scheme (DS).  The presentation made to the MPEG Multimedia Description Schemes (MDS) Group should identify sufficient reason and evidence for needing the DS.  The proposal should also provide justification of the specific structure of the DS and its relationship with other DS.

The subsequent review and evaluation of the proposal by the MDS Group may result in the establishment of a Validation Experiment (VE) or Core Experiment (CE) in order to better study the DS.

If the DS successfully progresses to the MPEG-7 Experimental Model (XM) or the MPEG-7 Working Draft (WD) stages, software will need to be provided by the proponents of the DS to allow its implementation.

*How are Multimedia Description Schemes validated by MPEG?*

The Multimedia Description Schemes are validated using one of two procedures depending on the type of Description Scheme (DS). First, the specific DSs related to content management, creation, production and usage go through a "harmonization" process in which the important concepts for description are identified and verified in other audio-visual data standards and existing practices.

All other remaining DSs go though an experimentation and verification process in which multiple companies implement the DSs in the context of audio-visual applications and demonstrate the usefulness and effectiveness of the DSs.

In both cases, ultimately, it is the consensus of the Multimedia Description Schemes (MDS) Group that determines the sufficiency of the harmonization, validation, and promotion of the DSs to the MPEG-7 MDS XM.

*What is a Multimedia Description Scheme Core Experiment?*

A Multimedia Description Scheme Core Experiment (CE) is a particular kind of experiment in which competing proposals for Description Schemes (DSs) are compared. The situation may arise when a new proposal is made with the intention of revising, replacing, or removing a DS that is in the MPEG-7 Multimedia Description Schemes (MDS) XM. A CE represents a head-to-head competition in which the alternative proposals are evaluated based on the comparative advantages.

*How can I join the Multimedia Description Schemes (MDS) Group?*

If you are a member of MPEG, you may join the Multimedia Description Schemes (MDS) Group by participating its meetings which are held at the MPEG meetings.

If you have any further questions, you may send email to the MDS Group reflector at MPEG-7-DS@advent.ee.columbia.edu. If you are not yet a member of MPEG, please see find information about joining at http://mpeg.tilab.com/ how_to_join.htm .

### XM software implementation FAQs

*Where can I get the XM source code? How can I get the source code?*

The XM-source code is located at cvshost.krdl.org.sg. The archive is managed by CVS. To get the source code do the following steps:

- Login command: cvs -d :pserver:cvs@cvshost.krdl.org.sg:/home/mpeg7-xm login

- Checkout command: cvs -d :pserver:cvs@cvshost.krdl.org.sg:/home/mpeg7-xm co newsrc

- Update command (updates your copy from the CVS server): cvs -d :pserver:cvs@cvshost.krdl.org.sg:/home/mpeg7-xm update newsrc

- Check in command: NEVER check in by yourself!! Always submit your code to S_Herrmann@lis.e-technik.tu-muenchen.de or to the AHG reflector

From time to time a snapshot is taken from the CVS server. This snapshot is available from a web server:

http://www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/e_mpeg7.html

The code on the web server does not provide the current revisions from the CVS.

*Which libraries are needed to compile the XM?*

There are three libraries required to run the XM with all functionality. These libraries are the XML parser, and the loaders for the media data (ImageMagick for visual, and AFSP for audio). The libraries required are also available from the Web server.

http://www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/e_mpeg7.html

*Are there templates for writing new modules for the XM?*

Yes, but only for the descriptor and description scheme components. The templates are the DummyType and the DSDummyType archives. These archives are available from the web server:

http://www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/e_mpeg7.html

*Where do I find information on how to compile the XM?*

Compilation instructions are available in the Readme.txt file comming with the XM source code.

*Which OS are supported for the compilation of the XM?*

In principle you can compile the XM on many OSs because the XM does not make use of OS specific functions. On the other hand, the XM requires external libraries, which are not part of the Reference Software. It is not garanteed that these libraries are avaliable on your desired platform.

However, the XM software is developed majorly on two paltforms: Linux and Win32. On the Linux side (and other Unix Systems) the compialtion process is supported by a Makefile.which is compatible to Gnu Make. The used C++ compiler requires supprt of namespaces. An applicable compiler is the Gnu C++ Compiler from version 2.9X.

For compilation on windows a "dsp" file is provided. As on Unix, the compiler needs to support namspaces, like ,e.g., Visual C++ 6.0.

*Is there some documentation about applications of the Ds/DSs implemented in the XM?*

Yes, the XM source code includes a directory named Doc. Here you can find three Subdirectories, one for Video, one for Audio, and one for MDS. All this directories include raw documentation files in plain ASCII format. Using the perlscript "format_all.pl" in the Doc directory you can compile the ASCII files to html pages. You can see the html pages by opening the file Doc/html/index.html in your web browser.

**MPEG-7 Conformance FAQs**

Editor's Note: currently there are no MPEG-7 Conformance FAQs.

**MPEG-7 Extraction and Use of Descriptions FAQs**

Editor's Note: currently there are no MPEG-7 Extraction and Use of Descriptions FAQs.